МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

3 | 1985

ISSN 0233-4844

Персональная ЭВМ «Ириша» предназначена для кабинетов информатики в общеобразовательных школах, ПТУ и техникумах

«Е-практикум» — диалоговая система ввода, редактирования и выполнения программ, задач и упражнений к учебнику информатики

Диспетчер памяти микроЭВМ «Электроника МС1213» позволяет адресовать до 256 Кбайт ОЗУ и реализовать мультипрограммный режим работы

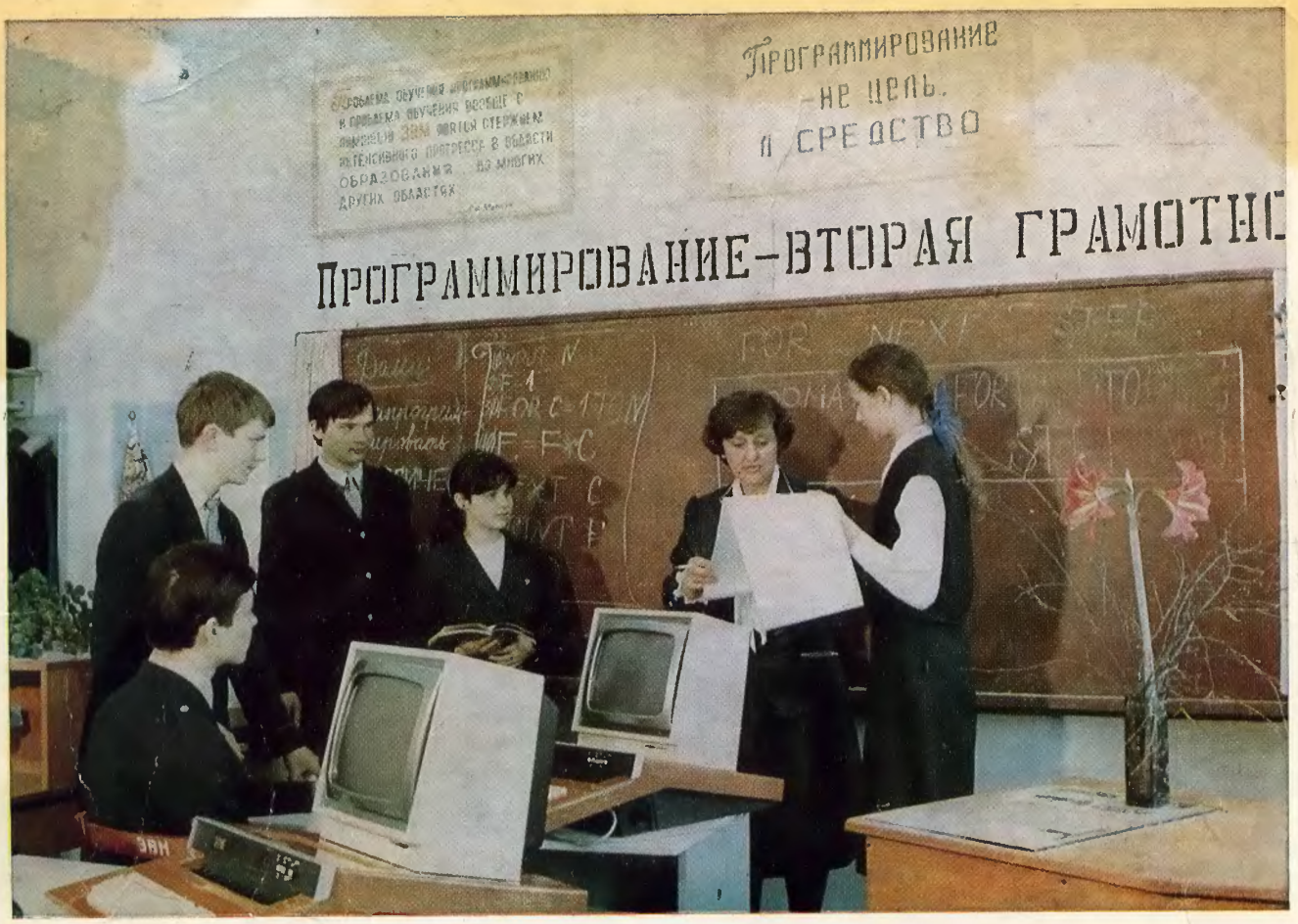
Постоянные запоминающие устройства (ПЗУ): физические основы занесения информации в ПЗУ; методика записи; программаторы

Схема формирования видеосигнала для телевизора, используемого в качестве дисплея бытовой персональной ЭВМ

Профессиональные применения персональных ЭВМ: обработка текстов, САПР, АРМ программиста, сбор и анализ измерительной информации

Компьютерные игры — важное средство расширения масштабов внедрения вычислительной техники





В 1975 г., поддержав инициативу школы по расширению компьютерного образования учащихся, академик Г. И. Марчук (в то время — директор ВЦ СО АН СССР) предложил установить в школе два терминала от ЭВМ, находящейся на территории ВЦ. Научное руководство работой возглавил академик А. П. Ершов (зав. отделом того же института). Тем самым работа получила направление не только учебно-практическое, но и научно-исследовательское.

Все три задачи, сформулированные в реформе школьного образования нашли отражение в опыте нашей школы:

— обучение курсу «Основы программирования и ЭВМ» («Школьная информатика») в рамках трудовой специализации учащихся старших классов (общий объем — 420 учебных часов, включая летнюю трудовую практику учащихся);

— профессиональная ориентация школьников в областях применения ЭВМ (прикладное программирование) через внеклассную работу учащихся (кабинет с «открытой дверью»);

— использование ЭВМ как технического средства обучения различным школьным дисциплинам.

Решению проблем школьной компьютеризации способствуют свободный доступ к ЭВМ через школьные терминалы в любое удобное для учащегося и педагога время, педагогический сервис, ставший возможным благодаря техническому, математическому и учебному обеспечениям вычислительного комплекса Сибирского отделения АН СССР и, наконец, отлаженность и бесперебойность службы коллективного пользования ВЦ и ГПВЦ СО АН СССР (Вычислительный центр и Главный производственный ВЦ), для которых мы — равноправные абоненты. К чести специалистов ГПВЦ, в ведении которых технические поддержание кабинета, неполадки терминального оборудования устраняются практически сразу после нашего вызова. Тем самым школа избавлена от необходимости содержать в штате не свойственных ей специалистов (инженеров-электронщиков).

Из педагогических достоинств матобеспечения выделим следующие:

— датчик случайных чисел обеспечивает автоматическую генерацию заданий, избавляя педагога от необходимости составления многовариантных работ: каждый учащийся получает от ЭВМ свой набор задач, примеров, вариантов игры и пр.;

— датчик времени позволяет оптимизировать обучение по индивидуальному темпу ребенка: педагог закладывает в обучающую программу лишь средне-статистическое время выполнения заданий, а ЭВМ по ходу обучения может это время либо увеличить, если учащийся из «мямликов», либо уменьшить, если он «шустрик»; тем самым есть возможность не только учесть внутренние физиологические особенности учащегося, но и опосредованно тренировать его по важнейшим психофизиологическим параметрам (например, на концентрацию внимания).

Свыше 500 учащихся вышли из стен школы с квалификационным свидетельством «Программист», и не менее 80 % наших выпускников ежегодно избирают профессии, прямо связанные с использованием ЭВМ.

Работа школьного компьютерного кабинета экспонируется на ВДНХ СССР в павильоне «Народное образование» в тематической выставке «Электронно-вычислительная техника в народном образовании».

Н. Л. Садовская
(г. Новосибирск, ср. шк. № 130. НГУ)

Пролетарии всех стран, соединяйтесь!

ОРГАН ГОСУДАРСТВЕННОГО КОМИТЕТА СССР ПО НАУКЕ И ТЕХНИКЕ

Издается с 1984 года

МП МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ЧЕТЫРЕ РАЗА В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 3 | 1985 МОСКВА

СОДЕРЖАНИЕ

Ершов А. П. — Колонка редактора 2

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Лопатин В. С., Борисенков В. Д., Юрочкин А. Г., Скокленев А. Г., Баранов Н. Д. — Диспетчер памяти микроЭВМ «Электроника МС 1213» 3

ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

Барышников В. Н., Воронов М. А., Гиглавый А. В., Паначев Ф. И., Романов В. Ю., Титов О. Ф. — Персональная ЭВМ «Ириша» для кабинетов информатики и вычислительной техники 5
Громов Г. Р. — Профессиональные приложения персональных ЭВМ 9
Лысенко Е. Е. — Компьютерная игра с точки зрения психолога 15
Кочетков Г. Б. — Компьютерные игры: свет и тени 16

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Корчак А. Е. — Интерпретатор языка программирования Бейсик/Г для микроЭВМ 21

КАК УЧИТЬ ПРОГРАММИРОВАНИЮ

Лосев И. С. — Размышления об обучении программированию 24
Варсанофьев Д. В., Кушниренко А. Г., Лебедев Г. В. — Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники 27

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Сумин В. В., Кислицын А. Б., Воробьев В. И., Савин А. Г., — Простейшие программируемые измерительные приборы на базе микроЭВМ 33
Воронин В. И., Макаров К. В., Старшова В. А. — Контроллер клавиатуры на базе однокристальной микроЭВМ КМ1816ВЕ48 36
Преснухин Л. Н., Фролов Г. И., Куправа Т. А., Безобразов В. С., Шахнов В. А. — Учебный класс на основе диалоговых вычислительных комплексов 39
Семенов П. А. — Микроконтроллеры на базе БИС КР580 для микроЭВМ «Электроника 60» и «Электроника НЦ-80-01Д» 42
Тарасов Н. А. — Устройство обработки информации с малым энергопотреблением 46

УЧЕБНЫЙ ЦЕНТР

Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г. — Принципы организации и работы дисплеев на основе БИС КР580ВГ75 51
Зеленко Г. В. — Дисплей для бытовой персональной ЭВМ 60
Буленков А. М., Буробин Н. В., Титчев Н. И. — Организация совмещенного ввода с перфолент в микроЭВМ «Электроника 60М» 70

УЧИСЬ РАБОТАТЬ С ПЗУ

Щербаков О. А. — Физические основы записи информации в ПЗУ . . . 72
Дианов А. П., Щелкунов Н. Н. — Методика программирования микросхем ПЗУ 75
Дианов А. П., Щелкунов Н. Н. — Модуль программирования микросхем ПЗУ 80
Лукьянов Д. А. — Схемотехника универсальных программаторов ПЗУ . . . 84
Барабанов А. Б., Турунов Н. Г., — Клавишное устройство ввода информации . 88
Справочная информация 91
Рефераты статей 95

© Всесоюзный центр информации по оборудованию ГКНТ СССР

Главный редактор
А. П. ЕРШОВ

Редакционная коллегия:

А. Г. Алексенко
В. В. Бойко
В. М. Брябрин
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. Смаков
В. И. Скурихин
В. Б. Смолов
Ю. М. Соломенцев
Н. Н. Шереметьевский

Технический редактор
Л. А. Горшкова
Корректор Г. Г. Казакова

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

Сдано в набор 01.08.85
Подписано к печати 18.09.85 Т-11508
Формат 84×108¹/₁₆. Бумага № 1.
Печать высокая. Усл. печ. л. 10,08.
Уч.-изд. л. 13,64. Тираж 21 600.
Заказ № 271 Цена 1 р. 10 к.

Орган Государственного комитета СССР по науке и технике.

Московская типография № 13 ПО «Периодика» ВО «Союзполиграфпром» Госкомиздата СССР

107005, Москва, Б-5, Денисовский пер., дом 30

На 1-й странице обложки — Первое знакомство со школьной микроЭВМ «Ириша» (см. ст. В. Н. Барышникова и др.)

ОБ ОБЪЕКТНО-ОРИЕНТИРОВАННОМ ВЗАИМОДЕЙСТВИИ С ЭВМ

Большая часть целеустремленной практической деятельности человека носит объектно-ориентированный характер. За этим несколько мудреным выражением скрывается совершенно очевидная вещь: человек всегда работает с каким-то видимым реальным предметом или воздействует на столь же реальную обстановку, имея для этой цели орудия или другие средства прямого воздействия, результат применения которых воспринимается также непосредственно и явно. Именно так выполняется работа по созданию материальных изделий и операторская работа — основные виды трудовой деятельности.

В течение многих лет программирование и решение задач на ЭВМ выглядели как антиподы такой предметной деятельности. Всячески подчеркивался абстрактный, незримый, планирующий, предугадывающий, формальный характер программирования.

Тем более представляет очень большой интерес и заслуживает распространения опыт объектно-ориентированного взаимодействия с ЭВМ, ставший реальностью за последнее десятилетие. На «верхнем слое» применений ЭВМ предвестниками этой методологии стали операторские тренажеры и некоторые системы автоматизации проектирования, где необходимость визуальной имитации обстановки или материального изделия составляли саму суть применения ЭВМ. Эти системы были, однако, слишком уникальными и дорогими, чтобы оказать серьезное влияние на методологию. Дополнительный импульс пришел от коммерческого бума производства электронных игр, где быстро сложилась полезная для многих потенциальных применений технология сотворения и визуализации искусственных «миров», в которых действует играющий.

Существенной материальной предпосылкой к реализации объектно-ориентированного взаимодействия с ЭВМ стал переход от алфавитно-цифровых терминалов к графическим дисплеям, существенно расширившим возможности визуализации объектов.

Пожалуй, первой серьезной демонстрацией объектно-ориентированного взаимодействия стали системы работы с документами, в которых была реализована «метафора письменного стола», т. е. визуализация привычной обстановки для человека, работающего с бумагами.

Обнаружилось, что объектно-ориентированное взаимодействие позволяет по-новому и при этом весьма продуктивно организовать работу и с более абстрактными объектами — таблицами, матрицами и т. п. Примером являются программы работы с динамическими бланками типа Visi-Calc, которые, пользуясь огромной популярностью у пользователей, далеких от программирования, реализуют в то же время весьма абстрактную схему деятельности человека.

Традиционно, понятие объектно-ориентированного программирования связывается с языком Смолтолк и поддерживающим его опытом исследовательской лаборатории компании Ксерокс в Пало Альто (США). На самом же деле, объектно-ориентированное взаимодействие не столько связывается с каким-то конкретным языком или программной системой, сколько является новым универсальным методом взаимодействия с ЭВМ, который с разной степенью осмысления и проработки был найден и применен в последние годы во многих профессиональных группах в разных странах.

Заметим, что объектно-ориентированное взаимодействие не отрицает теоретической природы программирования. Благодаря удачной символике, развитым средствам визуализации и мощной операционной поддержке оно придает ему ту непринужденность, наглядность, аналитическую интуицию, которая стала возможной в математике после разработки современной символики теории функций и анализа, придавшей математическому рассуждению графическую наглядность.

Начиная с этого номера (см. статью о Е-практикуме), и в дальнейшем в журнале будут публиковаться материалы, отражающие опыт отечественных и зарубежных специалистов в развитии и применении методов объектно-ориентированного взаимодействия с ЭВМ.

А. Ершов

УДК 681.3

В. С. Лопатин, В. Д. Борисенков, А. Г. Юрочкин, А. Г. Скокленев, Н. Д. Баранов

ДИСПЕТЧЕР ПАМЯТИ МИКРОЭВМ «ЭЛЕКТРОНИКА МС 1213»

МикроЭВМ «Электроника МС 1213» с диспетчером памяти имеет возможность адресации к памяти объемом до 256 Кбайт и располагает средствами для реализации мультипрограммного режима работы, т. е. обладает достоинствами мини-ЭВМ типа СМ-4, «Электроника 100/25» при сохранении преимуществ, которые дает микропроцессорная техника. Она программно совместима с указанными моделями мини-ЭВМ, а для достижения аппаратной совместимости располагает адаптером для связи с периферийными устройствами, имеющими интерфейс «Общая шина»*.

МикроЭВМ представляет собой набор функционально законченных устройств (модулей), объединенных с помощью магистрали, реализованной на основе межмодульного параллельного интерфейса (МПИ). МикроЭВМ не имеет отдельного пульта управления. Его функции может выполнять алфавитно-цифровое устройство ввода-вывода, способное принимать и передавать символы в кодах КОИ-7 (пультовой терминал). В режиме пультового терминала микроЭВМ может выполнять 13 команд, реализующих функции отладчика программы.

МикроЭВМ выполняет три типа операций обмена данными через МПИ: программный обмен, обмен в режиме передачи управления магистралью (режим прямого доступа к памяти) и режиме прерывания программы. Для обеспечения адресации к расширенному до 256 Кбайт адресному пространству дополнительно используются две адресные линии.

Центральный процессор (рис. 1, 2) микроЭВМ управляет предоставлением магистрали при совместном ее использовании внешними устройствами и выполняет безадресные, одноадресные и двухадресные команды, команды расширенной арифметики и

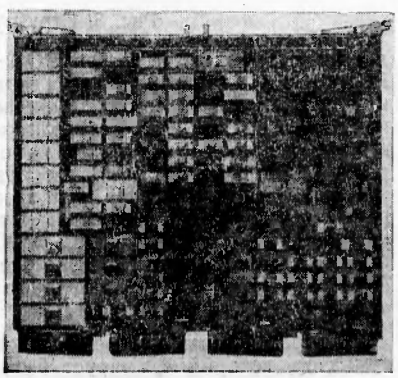


Рис. 1. Плата центрального процессора микроЭВМ «Электроника МС 1213»

плавающей запятой, может обрабатывать 8- и 16-разрядные слова.

Возможность использования восьми методов адресации позволяет проводить эффективную обработку данных, представленных отдельными словами или массивами с различной организацией и размещаемых в любых ячейках памяти. Предусмотрены средства обнаружения ошибок, нарушения нижней границы стека, нарушения четности адреса, ошибок четности оперативной памяти. Появление каждой из ошибок приводит к прерыванию программы.

Контроль нижней границы стека служит для защиты размещенной в памяти области векторов от чрезмерного расширения области стека. Для управления размерами защищенной зоны имеется регистр границы стека (РГС). Проверка адреса на четность производится при обращении к каждому слову данных, которые должны адресоваться по четному адресу.

В микроЭВМ используется память с контролем по четности, который разрешается программно, установкой

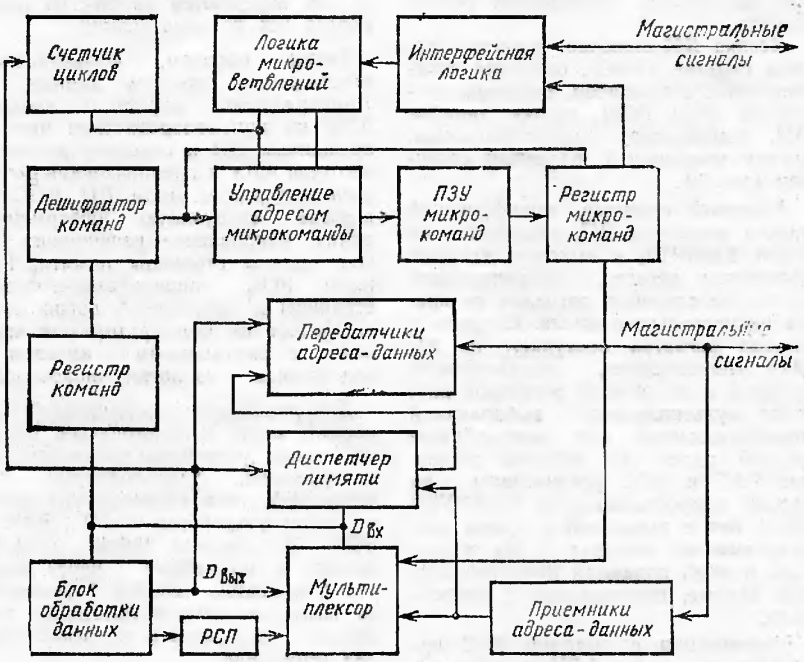


Рис. 2. Структурная схема центрального процессора

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

разряда регистра, расположенного на запоминающем устройстве (при записи данных каждый байт проверяется, и в случае ошибки на процессор поступает сигнал, вызывающий прерывание по ошибке четности).

Доступ к памяти объемом свыше 64 Кбайт производится под управлением диспетчера памяти (ДП). Если функционирование ДП разрешено, то виртуальное адресное пространство микроЭВМ, т. е. пространство, определяемое разрядностью процессора, разбивается на восемь страниц по 8 Кбайт, каждая из которых может независимо переадресовываться в любое место всего объема физической памяти. Обращение к страницам памяти управляется двумя наборами регистров, соответствующими системному и пользовательскому режимам работы процессора. Системный режим используется для работы управляющих программ, а пользовательский — для работы программ пользователя. Это обеспечивает дополнительную возможность защиты памяти от несанкционированного доступа.

Переадресация производится путем сложения виртуального адреса с константой, записанной в соответствующий регистр адреса страницы и задающей начальный (базовый) адрес страницы в физической памяти. ДП имеет средства защиты памяти, позволяющие запрещать доступ к памяти для чтения или записи. Это особенно важно при одновременной работе нескольких программ в режиме разделения времени. При попытке обращения к защищенной области памяти ДП прерывает выполнение программы, нарушившей режим защиты.

Логика ДП включает регистры адреса страниц (РАС), описания страниц (РОС), сумматор, регистры состояния (РС2, РС0), логику ошибок ДП, компаратор, мультиплексоры, логику прерываний, адресный селектор (рис. 3).

Адресный селектор, дешифрующий адреса регистров ДП, реализован на ПЛМ К556РТ2, к выходам которой подключен регистр, запоминающий состояние адресных сигналов на время магистрального цикла. Сигналы с выхода регистра поступают на логику чтения-записи, управляющую записью и чтением из регистров ДП, и на мультиплексор, выбирающий преобразованный или непреобразованный адрес. 16 наборов регистров РАС и РОС реализованы на восьми микросхемах ОЗУ К531РУ8П 16×4 бит с инверсией и тремя состояниями на выходах. На входы РАС и РОС подаются инвертированные данные, принимаемые с магистрали.

Информация с выходов РАС поступает на входы АЛУ и сумматора по мультиплексной шине. По той же шине на входы АЛУ передаются

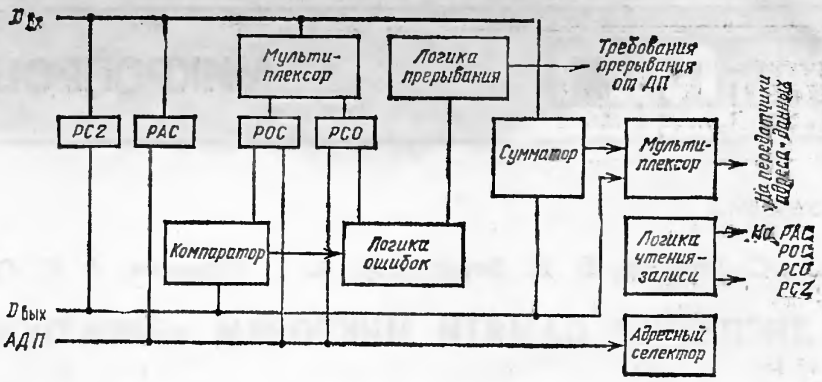


Рис. 3. Структура диспетчера памяти

данные с выходов регистров РОС и РС0, РС2 и мультиплексора входных данных АЛУ. Номер выбираемой пары РАС и РОС определяется номером страницы памяти, режимом работы ЦП, либо 8, 3, 2, 1 разрядами виртуального адреса при программном обращении к ним. При программном чтении из РАС, РОС, РС0, РС2 данные передаются на вход АЛУ по внутренней шине данных, минуя системную магистраль.

Для преобразования адреса служит сумматор, который суммирует 6 ... 12 разряды виртуального адреса, поступающего с АЛУ, с константой переадресации, содержащейся в 11 ... 00 разрядах РАС. Физический адрес с выходов сумматора поступает на вход мультиплексора и в адресной части магистрального цикла передается на 17 ... 06 линии адреса. Младшие шесть разрядов виртуального адреса передаются на 05 ... 00 линии адреса без преобразования.

Таким образом, мультиплексор обеспечивает передачу данных и 16-разрядного адреса с выходов АЛУ на приемопередатчики при запрещенном ДП и передачу данных с выходов АЛУ и преобразованного адреса при разрешенном ДП. РОС содержит следующую информацию: длину, направление расширения и код защиты страницы памяти. Разряды РОС, определяющие длину страницы и направление расширения, поступают на компаратор для сравнения с виртуальным адресом, а код защиты — на логику ошибок ДП.

Для реализации возможностей микроЭВМ могут использоваться дополнительные устройства ввода-вывода и контроллеры, разработанные для микроЭВМ типа «Электроника 60М», а также устройства СМ ЭВМ и ЭВМ «Электроника 100/25», подключаемые к микроЭВМ через адаптер с помощью кабелей. Имеющийся последовательный интерфейс позволяет подключать к микроЭВМ два терминала.

Устройства микроЭВМ «Электроника МС 1213» выполнены на микросхемах: К555, К531, К559, К1804, К556, К565. МикроЭВМ испытана в работе с дисковыми операционными системами ФОДОС, РАФОС, ТМОС, МДОСРВ и показала высокие эксплуатационные характеристики.

Статья поступила 11 декабря 1984 г.

БИНТИ № 2 (2195)

Фирма «Вотан» (Фремонт, шт. Калифорния) выпускает модуль синтезирования и распознавания речи, предназначенный для ввода и вывода информации в речевой форме при пользовании персональной ЭВМ «РС» фирмы «Интернэшнл бизнес мэшинз корпорейшн».

Речевая информация вводится в этот модуль через микрофон и с помощью специальных машинных программ преобразуется с частотой выборки около 8 кГц в цифровую форму. После тренировочного ввода в памяти модуля регистрируется лишь 3% информации о каждом слове, и в дальнейшем этого оказывается достаточно для распознавания слов. Синтезирование речи осуществляется в обратном порядке с выбором необходимых слов из рабочего словаря, хранящегося в памяти модуля. Модуль позволяет с помощью голосовой команды запускать в персональной ЭВМ исполнение прикладных программ.

В Японии для ЭВМ пятого поколения создается пищущая машинка с речевым вводом текстов, имеющая словарный запас 10 тыс. слов. Аналогичную пишущую машинку конструируют фирмы «Плесси» и «Шелл».

(США), т. 104, № 16, 1984 г., с. 3.

УДК 681.322.1

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

ПЕРСОНАЛЬНАЯ ЭВМ «ИРИША» ДЛЯ КАБИНЕТОВ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Прежде чем описывать персональную ЭВМ (ПЭВМ) «Ириша», необходимо понять, что такое компьютер для школы, для кабинетов информатики и вычислительной техники. Что должен уметь такой компьютер? Каким он должен быть по конструкции, возможностям, размерам и стоимости? С этими отправными вопросами неизбежно сталкивается конструктор, начиная разработку компьютеров этого класса. Здесь и возникает ряд трудностей, своего рода порочный круг: с одной стороны — для того чтобы составить программу машинного курса, очень хотелось бы знать возможности будущей школьной ЭВМ; с другой стороны — сами характеристики машины определяются по требованиям программы такого курса. Ошибки, промахи, недочеты в архитектуре, конструкции, возможностях программного обеспечения очень нежелательны по причине массовости такого устройства.

Немаловажны и социальные последствия этих промахов: школьный компьютер будет первым серьезным устройством, с которым столкнутся школьники, и нужно, чтобы они после своей первой встречи с компьютером поняли, что это их друг и верный помощник, а не очередное занудливое учебное пособие. Процесс формирования окончательного варианта школьного компьютера, видимо, будет состоять из ряда приближений. Сформулированы (по сегодняшним представлениям о курсе информатики в школе) требования к школьному компьютеру [1]. В ответ на них в ряде организаций разработаны конструкции и опытные образцы школьных ЭВМ, одна из которых — ПЭВМ «Ириша» (см. цветные иллюстрации на 1-й стр. обложки и на развороте вкладки).

Персональная ЭВМ «Ириша» в *минимальном комплекте* состоит из трех блоков: системного блока, телевизионного монитора и клавиатуры. В таком комплекте она должна, по замыслу авторов, существовать на рабочем месте учащегося в будущем кабинете информатики.

В *расширенном комплекте* ПЭВМ «Ириша» снабжается НГМД (150 и 203 мм) и принтером «Электроника УВВПЧ-30—004», подключаемым непосредственно к модулю процессора. В машину преподавателя можно установить дополнительную память (до 128 Кбайт), с помощью которой организуется электронный диск. Это существенно повышает скорость работы программ, требующих частого обмена информацией с диском. Расширенный комплект — вполне полноценный инструмент для подготовки программ. Кроме того, в локальной сети машин «Ириша», образующих кабинет информатики, машина преподавателя будет взаимодействовать с машинами на рабочих местах учащихся.

Безусловно, опытная эксплуатация подобных машин внесет свои коррективы в требования к ним, в их конструкцию. Предлагаемую конструкцию ПЭВМ следует рассматривать как некий промежуточный вариант, а сам компьютер, будучи выпущенным тиражом в сотни экземпляров, позволит скорректировать конструкцию и требования к новой машине будущего, тираж которой будет существенно больше. Возможно, часть ее узлов в дальнейшем будет исключена, а вместо них будут поставлены новые, необходимость в которых возникнет в процессе создания и отладки учебных программ.

Системный блок

В системном блоке находится сам компьютер и источник питания, а все остальное — это его периферия. Конструктивно системный блок представляет собой небольшую кассету, в которой может устанавливаться до 4 плат размером 150×230 мм (рис. 1).

В минимальном комплекте системный блок ПЭВМ «Ириша» состоит из двух модулей: модуля процессора и модуля телевизионного адаптера. Разделение модулей процессора и телевизионного адаптера позволяет достаточ-

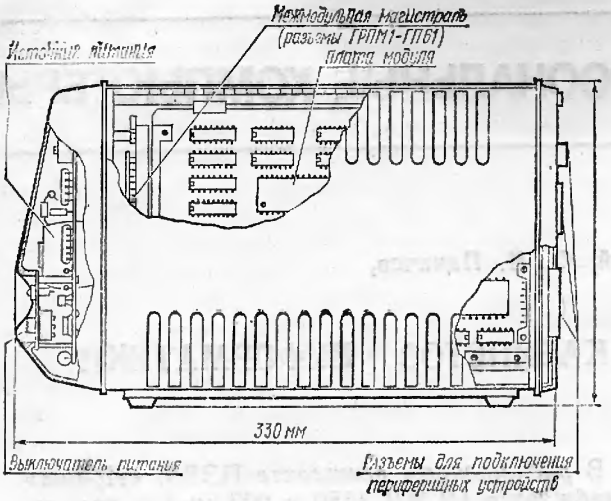


Рис. 1. Системный блок персональной ЭВМ «Ириша»

но гибко опробовать различные варианты процессоров при сохранении неизменного формата выдачи информации на экран монитора, и наоборот, а это очень важно на этапе отладки схемотехники компьютера.

Процессорный модуль (рис. 2) выполнен на базе микропроцессорного комплекта серии КР580. Выбор вызван доступностью комплекта, богатством и разнообразием уже существующего для него программного обеспечения (ОС-1800, микроДОС). И хотя МПК КР580, безусловно, не последнее достижение микроэлектроники, все же его система команд, видимо, наиболее подходит для построения 8-разрядных персональных ЭВМ.

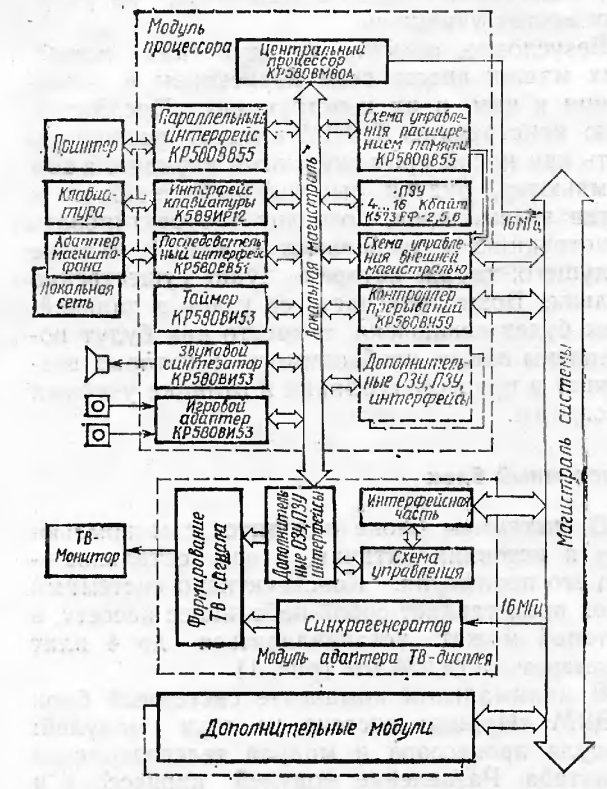


Рис. 2. Структурная схема персональной ЭВМ «Ириша»

Микропроцессор через локальную магистраль взаимодействует с узлами этой же платы: интерфейсом клавиатуры, многофункциональным параллельным интерфейсом для подключения принтера и других устройств, последовательным интерфейсом для работы с бытовым магнитофоном и локальной сетью, таймером, двухканальным звуковым синтезатором, двухканальным адаптером для работы с игровыми пультами, контроллером прерываний и узлом управления расширенной памятью. К этой же магистрали подключены и системные ПЗУ (4...16 Кбайт). Кроме того, на плате модуля процессора находятся аппаратные средства для работы с внешней магистралью и дополнительный разъем для технологического контроля и проверки. Через этот разъем к локальной магистрали процессора можно подключать дополнительные интерфейсные узлы, ОЗУ, ПЗУ и т. д.

Модуль телевизионного адаптера выполняет две функции: отображение информации на экране монитора и оперативной памяти компьютера. Он позволяет формировать на экране графическое изображение за счет сканирования определенной области памяти.

Модуль имеет три режима работы.

В монохромном режиме среднего разрешения модуль формирует одноцветное изображение (320×200 точек). Цвет засвечиваемых точек и фона может задаваться программно. Объем экранной памяти, необходимый для этого режима — 8 Кбайт.

В цветном графическом режиме и в режиме графики высокого разрешения объем экранной памяти — 16 Кбайт. В режиме цветной графики среднего разрешения каждая из засвечиваемых точек может быть окрашена в один из четырех цветов одной из двух заранее выбранных палитр. Графика высокого разрешения — одноцветная; рабочее поле 640×200 точек. Во втором и третьем режимах фон, на котором формируется изображение, может быть окрашен в один из 8 выбранных цветов. Эти два режима по организации вывода графического изображения аналогичны соответствующим режимам ПЭВМ IBM PC.

Общий объем ОЗУ модуля телевизионного адаптера — 64 Кбайт (8 БИС К565РУ5Д). Для регенерации изображения требуется 16 Кбайт (экранная память монитора), а остальные 48 Кбайт составляют оперативную память процессора.

Особенность ПЭВМ «Ириша» — отсутствие в ней специальных аппаратных средств для работы с символьной информацией. Текст на экране монитора формируется в графическом режиме программой, «зашитой» в ПЗУ процессора. Самая долгая операция обработки текста на экране (прокрутка) длится 0,2 с (практически незаметно для глаза).

Источник питания находится на передней панели системного блока под декоративной крышкой. Он преобразует входное постоянное напряжение 18...24 В в набор напряжений +5 В (5 А), +12 В (0,3 А) и —12 В (0,1 А), необходимых для работы системного блока ПЭВМ и клавиатуры.

По требованиям электробезопасности в школьных кабинетах не допускается разводка по столам учащихся напряжения свыше 42 В, и в действующем техническом задании на школьный компьютер это положение зафиксировано [1]. Но верхний предел 42 В оказался не слишком удобным для схемотехники источника питания; поэтому выбрано напряжение 18...24 В. Питание компьютеров (как и черно-белых телевизоров) на рабочих местах осуществляется от общего выпрямителя и разводится вместе с локальной сетью.

Механическая конструкция кассеты обеспечивает удобный доступ к элементам источника питания и одновременно надежно защищает ПЭВМ от чрезмерно любопытных молодых пользователей. Платы в кассете расположены вертикально, что обеспечивает достаточное охлаждение установленных на них элементов за счет естественной конвекции.

Устройство отображения

Устройство отображения — это черно-белый или цветной телевизор, переделанный в монитор. С увеличением выпуска школьных ЭВМ (большинству из них нужен монитор), промышленность должна выпускать простые мониторы, т. е. телевизоры без радиоканала. А пока можно использовать без переделки любой черно-белый телевизор, имеющий вход видеосигнала для подключения видеомагнитофона, либо доработанный до RGB-монитора цветной телевизор, либо специальный высококачественный монитор.

К ЭВМ «Ириша» черно-белый телевизор, имеющий вход видеосигнала («Вход ВИДЕО» на задней панели телевизора), подключается просто с помощью коаксиального кабеля. Если такого входа нет, то в большинстве серийных телевизоров сделать его несложно.

Работа с цветными телевизорами связана с организацией, кроме входа видеосигнала, входов для непосредственного управления яркостью лучей кинескопа. Использование антенных входов телевизора для соединения с компьютером оправдано только для бытовых ЭВМ, когда потребитель стремится расширить функции уже имеющегося телевизора. Это для него может оказаться более важным, чем потеря качества изображения, особенно в случае цветного телевизора. Устройство отображения на столе учащегося имеет только одну функцию — монитора ПЭВМ, в комплекте с которым они должны будут поставляться.

В настоящее время для ПЭВМ «Ириша» успешно используются черно-белые («Сапфир 401», «Юность 405», «Электроника 408Д»), а также цветные телевизоры («Юность Ц404», «Шилялис Ц410Д», «Рубин Ц202» и др.). Опытная эксплуатация показала, что оптимальный размер экрана черно-белого монитора на рабочем месте учащегося равен 23... 31 см, а цветного — не менее 32 см по диагонали. При меньшем размере экрана глаза очень устают. Телевизоры с размером экрана более 61 см по диагонали можно использовать в качестве демонстрационного монитора на рабочем месте преподавателя. ПЭВМ «Ириша» на экране телевизионного монитора формирует изображение (поле из 320 точек по горизонтали и 200 точек по вертикали). Это позволяет отображать текст, состоящий из 40 символов в строке, и иметь на экране 25 таких строк. С ПЭВМ «Ириша» могут работать мониторы и более высокого качества: графическая и текстовая информация может отображаться на поле, состоящем из 640 точек по горизонтали и 200 точек по вертикали. Это дает возможность работать с текстами по 80 символов в строке.

Клавиатура

Клавиатура ПЭВМ «Ириша» — контактная, построенная по схеме со сканированием матрицы кнопок. Она формирует на выходе коды символов согласно ГОСТ 13052—74 (КОИ-8). Выходной интерфейс клавиатуры — байтный, параллельный. Клавиатура выполнена в основном на микросхемах серии К561 и потребляет (только элементами индикации) при отсутствии передачи кода ток $\leqslant$40 мА, а в момент передачи — не более 100 мА. Она содержит 67 клавиш и имеет размеры 328$\times$ $\times$152$\times$30 мм. Вместо описанной клавиатуры можно использовать любую другую, совместимую по интерфейсным сигналам, например бо́льшую по габаритам клавиатуру 15ВВВ-97-006, обычно применяемую с дисплеем 15ИЭ-0-13.

Программное обеспечение

В настоящее время, когда программное обеспечение учебных курсов еще разрабатывается,

особенно важно предоставить в распоряжение пользователей наиболее широкий набор инструментальных средств программирования, в том числе: языков, средств обработки текстов, ведения баз данных (игровых и учебных).

В дисковом варианте ПЭВМ «Ириша» оснащена операционной системой, являющейся развитием ОС-1800 микроЭВМ СМ-1800. Это позволяет использовать многочисленные системные и прикладные программы, работающие под управлением ОС-1800: трансляторы с языков Бейсик, Фортран, Паскаль, СИ, макроассемблер, компоновщик и др. Ведется работа по переносу языка «Рапира» [2].

Каждый из указанных языков обладает уникальными свойствами, различаясь мощностью предоставляемых средств, простотой в изучении и использовании, эффективностью создаваемых программ. Выбор конкретного языка определится уровнем подготовки разработчика учебных программ и сложностью решаемой задачи.

Менее традиционно — применение в учебном процессе профессиональных средств работы с таблицами, позволяющих перейти от простой табличной модели (недалеко ушедшей по приемам работы с ней от калькулятора) к матричным моделям, алгоритмы которых могут быть достаточно сложными (оптимизация и т. п.). Пользовательский интерфейс остается простым — это «окно», перемещающееся по таблице памяти.

Очень важно использовать графические образы в учебном процессе. В настоящее время разрабатывается стандартный графический программный интерфейс пользователя ПЭВМ «Ириша».

По замыслу авторов, бездисковый вариант ПЭВМ «Ириша» в составе кабинета информатики и вычислительной техники должен иметь «зашитый» в ПЗУ Бейсик и через локальную сеть получать доступ к ресурсам дисковой машины преподавателя.

Применение

На сегодняшний день, когда мы находимся на этапе создания учебных программ, школьная ЭВМ также должна стать достаточно мощным программно-аппаратным инструментом для их разработки. Необходимо усовершенствовать методики преподавания различных дисциплин с использованием компьютера и создать соответствующие пакеты программ. Причем объем этого программного обеспечения столь велик, что не обойтись без привлечения к его разработке не только сил НИИ АПН и АН СССР, но и студентов педвузов, актива школьных преподавателей и учащихся. О том, что это реально, говорит опыт разработки языка «Рапира» в ВЦ СО АН СССР [2], зарубежный опыт. На наш взгляд, школе уже сегодня необходим учебно-производственный компьютер, т. е. профессиональный инструмент в специальном исполнении, на котором можно отрабатывать различные программы, методики и саму структуру компьютера в том числе. В качестве одной из моделей школьной ПЭВМ и в порядке эксперимента начата подготовка выпуска и внедрение опытных образцов ПЭВМ «Ириша» в системе образования Молдавской ССР.

Эти работы проводятся при активной поддержке ЦК КП Молдавской ССР силами промышленных предприятий, вузов г. Кишинева и республиканского института усовершенствования учителей в сотрудничестве с разработчиками кабинета учебной вычислительной техники на базе ПЭВМ «Ириша» — Химическим факультетом МГУ им. М. В. Ломоносова, ИПИ АН СССР и ВЦ СО АН СССР.

В республике предполагается начать работы по адаптации на ПЭВМ «Ириша» разработанного в ВЦ СО АН СССР пакета программного обеспечения «Школьница» и, в первую очередь, языка «Рапира». Ведутся также подготовительные работы по разработке версии программного обеспечения с использованием в качестве этнического — молдавского языка.

Учебный процесс — не единственная область применения «Ириши». Ее модули без изменений монтируются и в систему автоматизации научных исследований «Вариант-50» [3, 4], применяемую в физико-химических экспериментах. Замена микроЭВМ «Электроника ДЗ-28» в этой системе на ПЭВМ «Ириша» почти в два раза уменьшает стоимость и габаритные размеры и почти в три раза повышает ее производительность.

За справками обращаться по тел. 139-10-87 и 135-61-17.

ЛИТЕРАТУРА

1. Кабинет вычислительной техники всех типов средних учебных заведений / Под ред. С. Г. Шаповаленко.— М.: Ротапринт НИИШОТСО АПН СССР, 1985.

2. Звенигородский Г. А., Глаголева Н. Г., Земцов П. А., Налимов Е. В., Цикоза В. А. Программная система «Школьница» и ее реализация на персональных ЭВМ.— Микропроцессорные средства и системы, 1984, № 1, с. 50—55.

3. Романов Б. Ю., Барышников В. Н., Кулаков В. Б., Паначев Ф. И., Шаповалов В. Л. Набор функциональных модулей для построения систем автоматизации физико-химического эксперимента.— В кн.: Автоматизация научных исследований.— М.: Изд-во МГУ, 1984, с. 154—159.

4. Романов В. Ю., Барышников В. Н., Паначев Ф. И., Внуков П. В. Расширяемая система автоматизации простых экспериментов и приборов «Вариант-50».— В кн.: Информатика, вычислительная техника, автоматизация в науке и технике, народном хозяйстве. М.: ВИНИТИ, 1983, с. 104.

Статья поступила 10 июля 1985 г.

УДК 681.322—181.5+681.322.066+681.3 : 001.18

Г. Р. Громов

ПРОФЕССИОНАЛЬНЫЕ ПРИЛОЖЕНИЯ ПЕРСОНАЛЬНЫХ ЭВМ*

Введение

Как видно из рисунка и табл. 1, персональные ЭВМ являются в настоящее время классом машин, в значительной степени определяющим облик индустрии ЭВМ в целом.

Таблица 1

Структура мирового парка ЭВМ в 1984 г.

Класс ЭВМ	Мировой парк (шт.)*	Ведущие зарубежные фирмы	Доля ЭВМ данного класса в годовом объеме продаж (в процентах)
Супер-ЭВМ	100	CRAY RESEARCH CDC	2
Большие ЭВМ	100 000	IBM, «BUNCH» **	40
Мини-ЭВМ	1 млн.	DEC, H.-P., WANG, DATA GENERAL	15
Персональные компьютеры	20 млн.	IBM, Apple, TANDY, COMMODOR	43

* Цифры этой колонки дают приближенную оценку парка ЭВМ (с точностью до порядка);

** «BUNCH» — сложившаяся 15—20 лет назад аббревиатура «пятерки» крупнейших (после IBM) изготовителей больших ЭВМ в США: Burroughs, Univac (Sperry Corp.), NCR, CDC, Honywell.

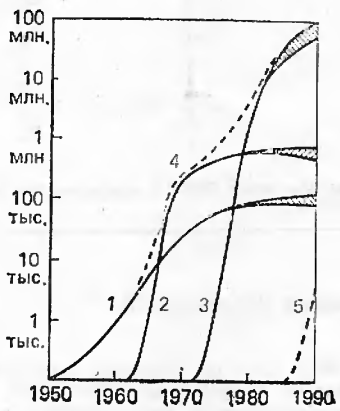
От 70 до 90 % суммарных расходов на персональные ЭВМ (ПЭВМ) идут непосредственно в профессиональные области их приложений: автоматизация учреждений; АРМ инженера, технолога, экономиста и т. д. В конце 1984 г. около 90 % всех ПЭВМ, производимых двумя крупнейшими фирмами этого сектора мировой индустрии ЭВМ (IBM, Apple Comp.) приобретались для профессионального использования.

По числу установленных в мире ПЭВМ так называемые «домашние» компьютеры (home computer), приобретаемые в основном для целей развлечения, и «профессиональные», используемые в трудовой деятельности, сопоставимы. Однако, если учесть, что стоимость профессиональной ПЭВМ составляет в среднем 1,5—5 тыс. долл., а «домашней» — в 10 раз меньше, то становится понятным, что общие расходы на миллионы ПЭВМ, используемых в сфере развлечений (ПЭВМ фирм «Tandy», «Atary» и др.), и миллионы профессиональных ПЭВМ (фирм IBM; Apple Comp, H.-P, DEC и др.), различаются весьма заметно: **основной объем средств, вкладываемых в сектор ПЭВМ индустрии ЭВМ, направляется непосредственно в сферу трудовой деятельности с целью повышения производительности труда **.**

Вместе с тем следует отметить, что миллионы наиболее дешевых «домашних» ПЭВМ типа, например, ZX 81 английской фирмы Sinclair играли до сих пор роль «букваря» в системе распространения компьютерной грамотности. Освоив простую в эксплуатации дешевую машину, многие из «вновь обращенных» пользователей средств вычислительной техники переходят затем к работе с более сложными и дорогими моделями профессиональных ПЭВМ. В 1981 г. Клайн Синклер обратился к жителям Англии с призывом: «Мы живем в век вычислительной техники. В ваших интересах уметь пользоваться ЭВМ». За последующие 3 года фирмой Sinclair было выпущено более 3 млн. «ПЭВМ-букварей» типа ZX 81 и Spectrum. По уровню компьютерной грамотности Англия стала за эти годы первой среди стран Западной Европы.

Общий характер использования «домашних» ПЭВМ отражает структура расходов их пользователей на приобретение пакетов прикладных программ. Из 1,3 млрд. долл., затраченных пользователями «домашних» ПЭВМ в 1984 г. на программное обеспечение: 500 млн. долл.— игровые программы; 300 млн. долл.— «деловое» (конторское) программное обеспечение (программы обработки текстов и т. д.); 200 млн.— обучающие программы. Таким образом, массовое использование даже наиболее дешевых «домашних» ПЭВМ, иногда называемых также «хобби-машинами» (hobby computers), носит далеко не только развлекательный, а в значительной степени деловой, прагматический характер *.

По ежегодному объему продаж профессиональных ПЭВМ с большим отрывом от ближайших конкурентов шли до начала 1985 г. первые две фирмы: IBM и Apple Comp. (см. табл. 2). Об особенностях архитектуры и некоторых технических характеристиках ПЭВМ фирмы Apple Comp. и первых моделей IBM PC см., например, [3, 4].



Динамика развития мирового парка ЭВМ [7, с. 52]: *1* — большие ЭВМ, *2* — мини-ЭВМ, *3* — персональные ЭВМ, *4* — суммарный парк универсальных ЭВМ, *5* — новый тип ЭВМ (образца 90-х годов)

* Статья представляет собой переработанный для журнала текст доклада, прочитанного автором 27.III.1985 г. в Институте прикладной математики им. М. В. Келдыша.

** Из общего объема продаж ПЭВМ, оцениваемого в пределах 20—30 млрд. долл., мировой объем продажи «домашних» компьютеров оценивался в 1984 г. на уровне 2—3 млрд. долл.

* При этом необходимо также учитывать, что значительная часть чисто развлекательных «игровых программ» выполняет для пользователей домашних ПЭВМ (нередко даже помимо их воли) функции первых разделов вводного курса компьютерной грамотности. О других аспектах влияния игровой компоненты ПЭВМ в том числе на процессы конструктивного творчества см. в [4, с. 56—69; 7, с. 124—127].

Цикличность развития индустрии ПЭВМ

К началу 1985 г., когда общее число фирм, производящих ПЭВМ, оценивалось уже на уровне 300, появились первые признаки того, что границы сектора персональных компьютеров мировой индустрии ЭВМ, наконец, обозначились. Как и всегда в таких ситуациях, немедленно возникла «обратная волна»: эпидемия банкротств, быстрое сокращение числа фирм, оперирующих в секторе ПЭВМ (по некоторым прогнозам еще до начала 1986 г. число их может сократиться с 300 до 70), спад мирового объема их производства. По мнению ряда ведущих экспертов [15], это обстоятельство является закономерным проявлением этапа экономической зрелости, в который вступает новая отрасль — индустрия ПЭВМ. Отмечается, что аналогичным образом в свое время формировались в США индустрия автомобилестроения (от десятков фирм — к «большой тройке») и многие другие отрасли.

Как и любая зрелая отрасль, индустрия ПЭВМ оказывается все более тесно связанной с самыми различными секторами хозяйственного механизма и начинает заметно сильнее реагировать на общую цикличность развития капиталистической экономики. В первой половине 1985 г. продолжающийся спад в электронной промышленности США, симптомы которого были зарегистрированы еще в 3-м квартале 1984 г., достиг уровня, когда начали «худеть» портфели заказов на ПЭВМ даже у фирм-лидеров: IBM и Apple Comp.

«Спасательный маневр», избранный в кризисной ситуации фирмой IBM, заключался в быстром перераспределении производственных мощностей для преимущественного выпуска двухпроцессорных машин класса «кентавр». Это ПЭВМ, встраиваемые в традиционные изделия мира больших машин, например, популярный дисплей 3270, который выпускается в одном корпусе с IBM PC и, соответственно, называется персональным компьютером IBM 3270/PC; мини-ЭВМ типа Series/1 в настольном исполнении, объединенная конструктивно с компьютерами IBM PC/AT или IBM PC/XT; выпускаются также настольные «большие» машины (совместимые с семействами IBM 4700 и IBM 370), представляющие собой, как правило, аппаратное расширение IBM PC/XT, например ПЭВМ типа IBM 4700 PC и IBM PC/370 [8—12].

Компания Apple Comp. не имела к началу кризиса 1985 г. того запаса устойчивости, который обеспечивал фирме IBM ее многомиллиардный парк больших ЭВМ, и, кроме того, руководством компании была в это же время допущена стратегическая ошибка — организован массовый выпуск профессиональных ПЭВМ типа Macintosh, не обеспеченных достаточным для успешного сбыта заделом программного обеспечения.

Разработчики машины Macintosh понимали, разумеется, уровень коммерческого риска, связанный с производством сравнительно дорогой профессиональной ПЭВМ, не обладающей программной совместимостью ни с их собственным семейством Apple II, ни с рядом машин IBM PC. Считалось, что по своим технико-экономическим характеристикам, а главное, достигнутому уровню «дружественности» человеко-машинного интерфейса, компьютер Macintosh окажется машиной значительно более высокого класса, чем любая из машин конкурирующего ряда IBM PC. Получив на свой рабочий стол мощный и одновременно простой, удобный в обращении рабочий инструмент обработки данных, десятки тысяч профессионалов из самых различных областей предметных приложений ПЭВМ немедленно начнут сами с энтузиазмом создавать для него прикладные программы, предполагали разработчики ПЭВМ Macintosh, а тем временем, как ожидалось, независимые программотехнические фирмы подготовят для коммерческой реализации базовые проблемно-ориентированные пакеты для новой машины.

Есть основания предполагать, что если бы рынок ПЭВМ продолжал расширяться с прежней скоростью (в IV квартале 1984 г. трехмесячный объем продаж ПЭВМ с маркой Apple Comp. достиг рекордного уровня в 700 млн. долл.), то и для ПЭВМ типа Macintosh, в рамках избранной стратегии, действительно могло оказаться на нем достаточно места. Однако к началу 1985 г. очередная волна кризисного цикла неожиданно для руководства фирмы Apple Comp. накрыла индустрию ЭВМ и, в том числе, сектор мини- и микроЭВМ, никогда ранее не знавший таких потрясений.

Согласно общей формуле рынка ПЭВМ: «программы продают машины» («software sell hardware»), при появлении признаков кризисного падения спроса на ПЭВМ в первую очередь начали останавливаться заводы по выпуску Macintosh — машины, не имеющей ранее накопленного задела программного обеспечения. Одновременно, хотя и по другим причинам, начал быстро падать спрос на ПЭВМ типа Apple II — до последнего времени весьма популярный действующий «памятник» первого поколения ПЭВМ.

В результате, первооткрыватель эры ПЭВМ в США — фирма Apple Comp. оказалась перед наиболее серьезными в ее истории, а по некоторым оценкам середины 1985 г., катастрофическими экономическими трудностями.

Таблица 2

Распределение мирового рынка профессиональных ПЭВМ в 1984 г.*

Фирма	Доля мирового рынка ПЭВМ, %
IBM	33
Apple	20
H.-P.	6
TANDY	5
DEC	3
NEC (Яп.)	3
Остальные	30

* Настольные и носимые ПЭВМ в диапазоне цен 1—6 тыс. долл.
Источник: Infocorp.

Автоматизация учреждений

До последнего времени наиболее массовой областью профессиональных приложений ЭВМ были отдельные подсистемы автоматизации учреждений с относительно медленной, но устойчивой тенденцией к их интеграции.

В первую очередь на базе ПЭВМ создавались компьютеризованные системы обработки текстов (word processing), а также системы ввода и обработки информации в табличной форме, сравнительно просто адаптируемые к структуре различного рода финансовых документов (ведомостей, бланков и т. д.). Наиболее популярным пакетом обработки текстов стал ППП Word Star, а из систем обработки табличной информации Visi-Calc. Общий тираж дискетт с пакетом World Star превысил недавно миллион экземпляров и Word Star оказался, таким образом, первым цифровым «золотым диском».

Несмотря на все более широкое использование в профессиональных ПЭВМ «твердых» дисков герметичной конструкции типа Винчестер, наиболее характерным устройством внешней памяти для профессиональных ПЭВМ все еще остается гибкий диск. Для большинства машин этого типа он является основным ВЗУ, а в случае, когда ПЭВМ оснащена, кроме того, диском типа Винчестер, выполняет по отношению к нему функции дублирования (back up) для обмена носителями информации между машинами. Общий тираж дискетт — носителей информации для гибких дисков — оценивается в настоящее время на уровне миллиарда экземпляров. Если принять во внимание, что одна дискетта средних размеров (например, 5¼-люймовая дискетта для ПЭВМ типа Apple-IIe) позволяет хранить несколько десятков страниц машинописного текста (т. е. содержимое средних размеров канцелярской папки), то можно оценить масштабы и характер качественных сдвигов, которые несет ПЭВМ в сферу документооборота промышленно развитых стран. Согласно официальным данным, в США, например, ежегодно правительственные службы должны обрабатывать около 300 млрд. ед. различного рода документов, поступающих из штатов, местных правительственных учреждений и от частных лиц. Этот бумажный конвейер обходился США к началу 80-х годов в 150 млрд. долл. ежегодных расходов, а единственная надежда приостановить быстрый рост финансовой тяжести этих бумажных потоков связывалась тогда с переводом большей части документооборота на машинные носители информации. К середине 80-х годов таким массовым машинным носителем информации, представляющим собой первую экономичную альтернативу традиционной пачке бумаг в картонном переплете (канцелярской папке), оказались гибкие диски.

Ленты и пакеты твердых дисков остаются пока основным средством централизованного хранения информации, в то время как на гибкие диски переводится все

более заметная часть нижнего уровня информационного потока, который циркулирует в «терминальных» первичных звеньях хозяйственного механизма. В основном, это информация, хранившаяся до сих пор на кабинетных полках с бумажными документами и в рабочих столах непосредственных исполнителей.

Известно, что на каждый следующий, более высокий уровень структурной иерархии учреждений, передается менее 10—20 % от всего объема циркулирующей на данном уровне информации. Гибкие диски, используемые на самых нижних уровнях иерархии информационных потоков, должны будут принять на себя, соответственно, подавляющую часть всего объема хранимой на машинных носителях информации.

Интегрирование отдельных подсистем автоматизации учреждений в «электронную контору» или «офис будущего» (office of future) шло до сих пор по двум взаимосвязанным направлениям. Первое было связано с созданием локальных сетей из группы ПЭВМ; подключением индивидуальных рабочих станций на базе ПЭВМ к большим ЭВМ, например в режиме эмуляции терминала и т. д. Второе — объединение пакетов программ, реализующих отдельные функции работы с документами в интегрированные программные системы типа, например, Lotus 1-2-3, или более поздняя разработка той же фирмы — пакет Symphony [12]. Пакет Symphony предоставляет пользователю ПЭВМ следующие функции: обработка текстов; работа с табличными данными; база данных для хранения документов и продуктов их обработки; режим связи с другими машинами («электронная почта»); средства экранной графики и, наконец, «многооконное взаимодействие».

Последний из упомянутых режимов реализует так называемую метафору рабочего стола [16, 5, 6], когда поверхность экрана ПЭВМ имитирует поверхность письменного стола с ворохом рабочих бумаг на нем. На экране можно отображать одновременно несколько документов в различных маштабах, передвигать их независимо, выбирать нужный в данный момент или его фрагмент и т. д.

База данных пакета Simphony позволяет хранить до 8 тыс. документов, а механизм работы с «динамическими таблицами» (spread sheet) — формировать и обрабатывать «поля данных» размерами до 8192 строк и 256 столбцов.

Научно-технические приложения

Научно-технические приложения потребляют относительно небольшую, но быстрорастущую часть общего тиража ПЭВМ. Для машины с маркой «Apple» научно-технические приложения оценивались к середине 80-х годов на уровне от одной пятой до четверти всего объема продаж профессиональных ПЭВМ этой фирмы. Куда конкретно идут эти машины?

Измерительные приборы

Вместе с одноплатным модулем многоканального АЦП такие ПЭВМ используются в качестве систем автоматизированного сбора данных, когда содержательная обработка измерительной информации должна производиться в непосредственной близости от источника сигналов. При этом техника «многооконного» отображения позволяет одновременно видеть на экране ПЭВМ исходный сигнал и результаты его обработки по различным алгоритмам, сопоставлять независимо отдельные фрагменты одного процесса и их взаимодействие и т. д.

Созданы и коммерчески тиражируются пакеты прикладных программ, которые позволяют использовать популярные модели ПЭВМ в качестве многоканальных осциллографов с памятью, статистических анализаторов и т. д.

Контроллеры

В цехах промышленных предприятий ПЭВМ используются в качестве информационного ядра систем управления станками и оборудованием технологическими процессами; в лабораториях — в качестве интеллектуального пульта стойки управления научной аппаратурой.

Самой массовой областью приложений встраиваемых ПЭВМ в ближайшее время, видимо, будет использование их в качестве простого и широко доступного производственному персоналу универсального контроллера для управления станками и технологическими комплексами, требующими быстрой замены или перестройки режима функционирования. Например, согласно известным оценкам, к 1990 г. менее трети всех действующих в цехах американских предприятий контроллеров производственного оборудования еще можно будет отнести к классу доминирующих сегодня «черных ящиков» — приборов с жестко запаянной логикой управления; 20 % — будут составлять простейшие адаптивные контроллеры»; 50 % — универсальные перепрограммируемые контроллеры. Именно эта последняя «половина» всего многомиллионного парка производственных контроллеров — основная область профессиональных приложений «встраиваемых ПЭВМ» (OEM — personal computers).

Инструментальные средства

Годовой объем мирового производства однокристальных микроЭВМ, встраиваемых в автоматизированные устройства приборов, машин и агрегатов к 1990 г. достигнет 1 млрд. экз. Но массовое внедрение их в народное хозяйство до сих пор сдерживается в значительной степени трудностями разработки программного обеспечения и отладки систем непосредственно на объекте автоматизации. Во второй половине 80-х годов эти трудности резко возрастают в связи с быстрым расширением номенклатуры автоматизируемых систем, использующих микроЭВМ в режиме реального времени: производственное оборудование, бытовая техника, научная аппаратура, медицинские приборы и т. д. До начала 80-х годов наиболее массовым инструментальным средством разработки программ для встраиваемых микроЭВМ и микропроцессоров (MDS-microcomputer development system) были MDS типа Intellec фирмы Intel. К 1984 г. мировой парк этого типа MDS оценивался в 65 тыс. действующих систем. Общее же число автоматизированных рабочих мест для инженеров-разработчиков программного обеспечения встраиваемых микропроцессорных систем оценивалось к этому времени на уровне 100 тыс. Пакеты прикладных программ, предназначенные для использования ПЭВМ в качестве универсальных кросс-систем, параметрически ориентируемых на разработку программ различных микроЭВМ, а также ряд одноплатных модулей для создания на базе ПЭВМ различных MDS с внутрисхемной эмуляцией, позволят во второй половине 80-х годов более чем на порядок расширить круг инженеров-разработчиков, имеющих личные автоматизированные рабочие места для создания программного обеспечения систем управления оборудованием на базе микроЭВМ.

САПР и супер-микро

До последнего времени задачи автоматизации инженерного проектирования в промышленности, строительстве и других областях традиционного инженерного творчества решались на машинах классов мини и супер-мини (например, для ЭВМ фирмы DEC это машины уровня PDP-11—70, VAX-780). Большие машины использовались заметно реже из-за необходимости существенно

Таблица 3

Рост общего парка САПР в США (в шт.)

Область использования	1981 г.	1985 г.
Проектирование механических устройств	1744	9 000
Электротехника и электроника	1620	3 700
Строительство и архитектура	630	2 900
Картография и другие области	568	2 400
Всего действующих систем	4562	18 000
Суммарная стоимость парка (млрд. долл.)	0,765	2,8

Источник: Predicasts Inc.

интерактивного режима работы, а ПЭВМ на таких задачах не хватало вычислительных ресурсов. Поэтому, несмотря на острую потребность растущего инженерного корпуса промышленно развитых стран в системах автоматизации проектирования (САПР), общий парк таких систем к середине 80-х годов все еще был далеко не достаточным, чтобы реально влиять на темпы роста производительности труда инженеров. Например, сотни тысяч инженеров США имели к этому времени около 20 тыс. САПР* (см. табл. 3). В основном первые поколения ПЭВМ сколько-нибудь заметно затронули лишь работу служащих учреждений.

Реальные же сдвиги в «механовооруженности» интеллектуального труда инженеров ожидаются с началом массового производства ПЭВМ третьего поколения. К ним относятся ПЭВМ с 32-разрядными микропроцессорами (8-разрядные ПЭВМ — I поколение; 16-разрядные — II поколение). Иногда упоминается и более общее для ПЭВМ третьего поколения определение — «трехмиллионная ПЭВМ». Это настольная машина с производительностью в миллион операций в секунду, ОЗУ емкостью в миллион слов и «миллионным дисплеем» — 1024×1024 точек. ПЭВМ с высокой производительностью, большой памятью и графикой высокого разрешения, удовлетворяющие указанным выше требованиям «трех миллионов», получили название «супермикро» или «32-битные рабочие станции». Типичными настольными компьютерами этого класса являются, например, рабочие станции DN460 и DN660 фирм Apollo Computer и HP 9000 фирмы Hewlett-Packard.

Индивидуальные рабочие станции Apollo построены на базе 32-разрядных секционированных (bit slice) микропроцессоров и объединяются в локальную сеть с общей виртуальной памятью. Это облегчает программисту организацию оперативного доступа независимо работающих проектировщиков к централизованной библиотеке прикладных программ, общей базе данных КБ, НИИ, отдела, где могут храниться справочные данные по сортаменту материалов, стандарты на узлы проектируемых изделий, файлы графических образцов конструкций и т. д. Каждая рабочая станция имеет до 4 Мбайт защищаемой избыточным кодированием (error corrected) оперативной памяти; 32-битную шину «процессор — ОЗУ». Операционная система локальной сети рабочих станций обеспечивает пользователям возможность параллельного исполнения до 24 процессов и предоставляет каждому из независимо исполняемых процессов до 256 Мбайт виртуальной памяти.

Дисплей компьютера DN460 монохромный, растровый: 1024×800 точек с поэлементным кодированием изображения (bit mapped) имеет встроенную память для хранения «картинок» на 128 Кбайт. В модели DN660 используется цветной дисплей с разрешением 1024×1024 и 2 Мбайтами встроенной памяти изображений.

Основная область приложений компьютеров DN460 и DN660 — задачи САПР в электротехнике и машиностроении, системы автоматизации сложных технологических процессов (АСУ ТП); инженерные задачи численного моделирования и другие инженерные и прикладные научные задачи. Языки программирования: ANSI Fortran 77, Pascal, «С».

Супер-микро HP 9000 выполнен на базе специально созданного фирмой Hewlett-Packard 32-разрядного микропроцессора «Focus» и имеет вычислительные ресурсы уровня мини-ЭВМ класса VAX (производительность — 1 млн. операций/с). Основная область приложений — САПР. Настольные компьютеры HP 9000, как и машины Apollo, объединяются в локальную сеть и имеют стандартные для научных приложений языки программирования (Fortran 77, Pascal, «С»).

Недавно созданная и относительно небольшая фирма Apollo оставалась последние годы в США лидером среди изготовителей профессиональных ПЭВМ третьего поколения, ориентированных на задачи САПР. Однако в ближайшее время в этот сектор рынка ЭВМ войдут ряд крупных компаний, включая IBM. Основной базой для создания массовых моделей супер-микро, видимо, станут 32-разрядные микроЭВМ типа M68020 (начало коммерческого производства — весна 1984 г.). Привлекают интерес разработчиков также микроЭВМ фирмы National Semiconductor 32032 (выпускаются с 1983 г.), упоминается в этом контексте и фирма Zilog с моделью Z80000. С мая 1982 г. идет коммерческое производство первого в мире 32-разрядного микрокомпьютера iAPX432 фирмы Intel. Вместе с сопроцессором плавающей арифметики супер-микроЭВМ на базе iAPX432 обеспечивает производительность на уровне 1млн. плавающих операций/с.

Отметим кратко основные технические характеристики явного пока фаворита в ожидаемой гонке за преобладание в секторе супер-микроЭВМ — микрокомпьютера M68020: 32-разрядная магистраль данных, непосредственно адресуемое ОЗУ — 4 Гбайта; производительность — 2—3 миллиона операций в секунду (тактовая частота 16,7 МГц). Питание осуществляется от одного источника 5 В, рассеиваемая мощность 2 Вт. Структурная сложность топологии кристалла оценивается в 200 тыс. эквивалентных транзисторов. Несмотря на то, что в этой существенно обновленной по отношению к M68000 и M6810 модели предусмотрено использование новых типов адресации, данных и команд, фирмой Motorola была обеспечена программная совместимость «снизу-вверх» микроЭВМ M68020 с базовой моделью популярного семейства второго поколения M68000.

Значительный интерес у научных работников, инженеров и изготовителей комплексного оборудования в области АСУ ТП и систем сбора данных вызвало появление на рынке профессиональных ПЭВМ компьютера IBM PC/AT, который может быть отнесен к промежуточному поколению «2, 5».

Фирма IBM назвала эту машину «ПЭВМ на базе передовой технологии» (AT — advanced technology). В компьютере IBM PC/AT используется 6-мегагерцевый

---

* Причем бо́льшая часть этого парка САПР используется пока для автоматизации лишь задач «верхнего слоя» — наиболее элементарных функций, выполняемых проектировщиком или инженером-конструктором. Так, например, 80 % всего парка САПР, применяемого в промышленности США, задействовано на задачах черчения — т. е. представляет собой, по существу, компьютеризованный «кульман». Однако следует отметить, что сама по себе возможность хранить чертежи и графические элементы конструкций в памяти ЭВМ уже заметно отражается на эффективности работы конструкторов. Согласно опубликованным в 1984 г. данным, даже в небольших КБ профиля, например, механообработки внедрение САПР позволяет снизить общие расходы на штат инженеров-конструкторов в пределах 30 % и повысить реальную отдачу каждого инженера на 60 % [14].

микропроцессор i80286, что позволило в два-три раза поднять производительность по отношению к модели IBM PC, выполненной на i8088. Как и микропроцессор i8086, модель 80286 является чисто 16-разрядной машиной с одинаковой (в битах) шириной внутренней и внешней шин данных. «Передовая» (по отношению к i8086) технология i80286 для пользователя ПЭВМ проявляется, в основном, следующими техническими характеристиками: виртуальная память до 1 Гбайта; два режима работы с операционной системой ПЭВМ:

1) «эмуляция» i8086, например для исполнения базовой для семейства IBM PC операционной системы PC — DOS (при этом размер непосредственно адресуемого ОЗУ оказывается ограничен 640 Кбайт);

2) работа с операционной системой XENIX, когда i80286 позволяет непосредственно адресовать до 16 Мбайт физической памяти (реально IBM PC/AT позволяет наращивать ОЗУ платами по 512 Кбайт до 3 Мбайт). Следует отметить, что с точки зрения работы ПЭВМ в контуре АСУ ТП, системах сбора и предварительной обработки в цеховых условиях измерительной информации и других системах реального времени представляют значительный интерес также следующие технические характеристики IBM PC/AT: 16 уровней прерываний; 7 независимых каналов прямого доступа в память.

ПЭВМ и ВЦКП

Через несколько лет после появления на рынке первых ПЭВМ руководители фирм-изготовителей столкнулись с неожиданными выводами служб анализа профессиональной структуры пользователей: заметную часть ПЭВМ класса Apple и IBM PC приобретали профессиональные программисты с десятилетним и более стажем работы на больших машинах. Чтобы попытаться прояснить эту внешне парадоксальную ситуацию, состоялось немало дискуссий * и был поставлен ряд экспериментов по прямому сопоставлению времени решения одних и тех же типовых задач на вычислительном центре коллективного пользования (ВЦКП) и за пультом ПЭВМ.

В качестве базы для сравнения в одном из таких экспериментов была выбрана рутинная для многих ВЦКП задача экономического моделирования [17]. Чистое процессорное время счета составило 4,5 мин для большой ЭВМ и измерялось часами для ПЭВМ (для сопоставления использовались специально написанные на языке близких версий программы, имеющие аналогичные по требуемым вычислительным ресурсам параметры). Однако полное время прохождения задания на ВЦКП (от загрузки условия до выдачи результатов моделирования на диск) составило 7 ч 28 мин — заметно больше, чем время получения пользователем результатов на каждой из участвовавших в эксперименте ПЭВМ (задачи пускались на ПЭВМ типа Apple II и IBM PC).

Сравнения, выполненные в большом числе организаций на различных задачах, показали, что ПЭВМ могли бы существенно разгрузить базовые машины ВЦКП от непроизводительной работы с «мелочью» минутного счета. Как правило, поток из большого числа относительно коротких по чистому процессорному времени задач составляет заметный фон почти на любом ВЦКП и стоит операционной системе базовой машины значительных накладных расходов, связанных с непроизводительной работой по непрерывному перераспределению общих вычислительных ресурсов ВЦКП. В этих условиях ПЭВМ оказались именно той технологической альтернативой, которая позволит преодолеть сложившееся на рубеже 80-х годов диалектическое противоречие в развитии ВЦКП: «пакетная обработка — режим разделения времени», решив его по принципу: «богу богово — кесарю кесарево». Эффективно решать на «больших машинах большие задачи» можно будет, предварительно отфильтровав на уровень ПЭВМ все остальное.

Первыми машинами этого «фильтрующего класса» оказались модели 3270/PC и XT/370, которыми фирма IBM расширила свое семейство ПЭВМ типа IBM PC. Это, видимо, пока единственные модели ПЭВМ, специально предназначенные для профессиональных пользователей больших ЭВМ.

Настольная машина IBM 3270/PC представляет собой стандартный для больших ЭВМ фирмы IBM терминал 3270 с встроенным в него дополнительным оборудованием, предоставляющим пользователю возможность исполнения в автономном режиме программ IBM PC. Сам по себе тот факт, что терминал, используемый для связи с большой ЭВМ, в состоянии автономно «решать задачи», должен, видимо, стимулировать пользователя к «плодотворным колебаниям» в минуты тягостного ожидания результата: стоит ли эта конкретная задача ресурсов большой ЭВМ или ее лучше переписать для более быстрой и комфортной отладки здесь же на рабочем месте в режиме IBM PC? При решении задач в режиме IBM PC машина работает только автономно и передача программ или файлов из большой машины в ПЭВМ (и обратно) штатными средствами терминала 3270 не поддерживается.

Вместе с тем, некоторые функциональные возможности IBM PC используются при работе 3270/PC в качестве терминала большой машины, например, «многооконное» отображение на экране дисплея одновременно нескольких сеансов диалога с большой ЭВМ.

Всего на экране ПЭВМ 3270/PC может быть одновременно до семи «окон». Четыре могут быть использованы для отображения независимо протекающих сеансов диалога с различными программами большой ЭВМ; одно — для работы с IBM PC; два — для «рабочих записей» пользователя (в этих двух «окнах» пользователь может готовить для отправки на другой терминал «письмо», использовать их в качестве буферного поля для визуального контроля обмена данными * между «окнами» и т. д.). Программа, с которой пользователь в данный момент работает с пульта 3270/PC, отображается в верхнем «окне», и может, при необходимости, последовательно вытеснять другие «окна», чтобы занять весь экран.

Следующим логическим шагом в попытках объединить в одну рабочую систему «коня и трепетную лань» — мощь вычислительных средств больших ЭВМ («байтодробилок») и функциональную гибкость микроЭВМ — явилось создание ПЭВМ типа XT/370. Архитектурно эта машина представляет собой последовательное развитие магистральной для ПЭВМ тенденции к объединению в одном корпусе ряда наиболее богатых по заделу прикладного программного обеспечения «разнопроцессорных» машин [4, 7]. Конструктивно каждое аппаратное расширение типовой конфигурации ПЭВМ для выполнения системы команд еще одной «конкурирующей модели» представляет собой отдельную плату (softcard — «карта» для расширения программного обеспечения). Например, в одно из свободных гнезд на шасси ПЭВМ типа Apple-II может быть вставлена «карта» с процессором Z80 («Z80 — softcard») для исполнения программ, написанных в наиболее популярной на уровне микроЭВМ ОС типа CP/M. Аналогичным образом пользователям ПЭВМ типа IBM PC/XT предоставлена возможность приобретать комплект из «трех карт», чтобы превратить, таким образом, свою маши-

* При обсуждении этого вопроса на конференции в Пущино, С. С. Лавров заметил, что разница в характере работы профессионального программиста за пультом ВЦКП или ПЭВМ «такая же, как между отдельной и коммунальной квартирами, даже при наилучших отношениях с соседями» [1, с. 7]. «Программистам надоело, — настаивал А. П. Ершов, — ходить без сапог и они хотят иметь у себя на столе хорошую ПЭВМ для профессиональной работы» [2, с. 10].

* Обмен данными возможен только между наблюдаемыми программами большой машины. Возможностей обмена: «большая машина — ПЭВМ» пользователю 3270/PC не предоставляется (во всяком случае, в первой версии этой машины).

ну в модель XT/370, способную исполнять значительную часть прикладных программ, написанных для больших ЭВМ ряда IBM 360/370.

ПЭВМ типа IBM PC/XT, укомплектованная тремя дополнительными платами (на одной расположены два микропроцессора M68000 и один i8087, обеспечивающие возможность исполнения большей части команд машин ряда IBM/370, вторая — полумегабайтная оперативная память, третья эмулирует работу терминала 3277 для связи с большой ЭВМ), предоставляет пользователю возможность, манипулируя ключом на пульте, получать в свое распоряжение одно из трех устройств: 1) обычную ПЭВМ типа IBM PC/XT; 2) «настольную» машину ряда IBM/370; 3) удаленный терминал (типа 3277) большой ЭВМ. Соединенная с одной из базовых машин ВЦКП настольная ЭВМ типа XT/370 дает возможность пользователю гибко маневрировать при решении различных задач ресурсами своей ЭВМ и общими для пользователей ВЦКП ресурсами большой машины.

Работа в режиме «настольной IBM 370» идет под управлением операционной системы VM/PC, команды которой аналогичны или идентичны командам VM/SP (версия 2) ряда больших машин IBM/370.

Пользователь «настольной IBM 370» имеет до 4 Мбайт виртуальной памяти. Внешняя память, в зависимости от числа используемых дисков типа Винчестер (1 или 2),—10 или 20 Мбайт.

Одно из основных преимуществ интеллектуального терминала на базе XT/370 перед другими типами ПЭВМ, используемыми для этой цели на ВЦКП,— возможность непосредственно обмениваться независимо исполняемыми программами и файлами между «настольной» и «большой» машинами.

Инфраструктура технического и программного обслуживания ПЭВМ

Экономические преимущества от внедрения ПЭВМ в рассмотренные выше области профессиональных приложений для любой отдельно взятой страны, начинающей процесс массовой компьютеризации народного хозяйства, являются потенциальными преимуществами.

Окупятся ли многомиллиардные расходы на закупку и производство сотен тысяч профессиональных ПЭВМ ожидаемым повышением производительности труда, зависит в первую очередь от того, окажется ли развертываемый парк ПЭВМ, по крайней мере, работоспособным.

В ряде стран накоплен в этом отношении определенный опыт, в том числе и негативный. Например, по результатам исследований, выполненных Индийским институтом технологии, подчеркивается «неэффективность пути промышленного копирования разработок фирм IBM и закупок новейших вычислительных средств в США, если страна-покупатель не обладает соответствующим уровнем развития инфраструктуры» [13].

На 9-м конгрессе Международной федерации по обработке данных (IFIP-83) отмечалось, что для эффективного использования микроЭВМ (как и других технологических достижений) в стране должна быть предварительно создана необходимая для их эксплуатации инфраструктура технического и программного обеспечения.

Во многих странах, где этот факт вовремя не был осознан, после приобретения ЭВМ пользователь не может, как правило, «получить ответ на важнейший из возникающих вопросов: где и как получить доступ к сети сервиса микроЭВМ?» [18, p. 472]. Подчеркивается, что до тех пор пока не будут преодолены трудности, связанные с организацией регулярного доступа пользователей микроЭВМ к источникам высококачественного программного обеспечения и экономически приемлемой системе технического обслуживания, возможность сколько-нибудь массовой компьютеризации остается весьма проблематичной [18, p. 473].

В отличие от всех остальных типов ЭВМ, которые предполагают существование в организации, приобретающей машину, бригады обслуживающего персонала (инженеры и техники-электронщики, программисты и т. д.), профессиональные ПЭВМ ориентированы в основном на «непрограммирующих профессионалов», которые, впервые оказавшись «один на один» с ЭВМ, знать ничего не хотят об ее обслуживании. Поэтому изготовители уже не могут больше перекладывать на плечи пользователей даже малую часть бремени поддержания жизнеспособности своего изделия. Конкурентная борьба на мировом рынке среди более чем 200 производителей ПЭВМ привела к беспрецедентной ситуации, когда в США, например, заказчики военной аппаратуры нередко указывают своим подрядчикам на массовые модели ПЭВМ как желаемый образец для задания требуемого им уровня надежности. Согласно одному из последних определений «ПЭВМ — это вычислительная машина с надежностью военной аппаратуры и ценой изделия бытовой электроники».

Следует отметить, что пользователю в определенных пределах почти безразлично, обеспечен ли достаточно высокий уровень эксплуатационной готовности его «личной ЭВМ» к практическому использованию только за счет достигнутой фирмой-изготовителем технологии производства высоконадежных, почти не требующих ремонта узлов или развитой сетью оперативного технического обслуживания ЭВМ у потребителя: например, диагностирование и замена в течение нескольких часов по телефонному вызову вышедшего из строя блока или узла ПЭВМ. Поэтому для достижения заданного (условиями конкуренции на мировом рынке) уровня коэффициента эксплуатационной готовности ПЭВМ каждая фирма находит свой экономически оптимальный для нее баланс средств, расходуемых непосредственно в производстве на разработку и реализацию конкретных технологических решений, повышающих эксплуатационную надежность ПЭВМ, с одной стороны, и на дальнейшее развитие фирменной системы послепродажного сервиса — с другой.

Например, на фирме IBM в настоящее время около 10 % общей численности сотрудников занято в службе сервиса (свыше 31 тыс. человек); на фирме DEC — около 25 %. В общем случае, чем выше надежность изделия, тем соответственно меньше нагрузка на сеть предприятий технического обслуживания, и наоборот. Однако в данном случае заметная разница в нагрузке на сеть сервиса определяется в значительной степени тем обстоятельством, что фирма DEC производит в основном мини-ЭВМ, используемые в контурах управления, где необходима большая «реактивность» службы сервиса.

Уровень сервиса и надежность — основные параметры, по которым идет сегодня на мировом промышленном рынке отбор ПЭВМ. Это обстоятельство в немалой степени объясняет тот внешне труднообъяснимый факт, что, например, на американском рынке ЭВМ пользователи явно предпочитают ПЭВМ типа IBM PC десяткам программно-совместимых моделей конкурирующих фирм, которые предлагают значительно более дешевые ПЭВМ с техническими характеристиками, как правило, не уступающими, а в ряде случаев и заметно превосходящими IBM PC.

Как показывает анализ, страна, начинающая процесс массовой компьютеризации, в первую очередь должна сосредоточить материально-технические ресурсы на решении главной задачи — создать и опережающими темпами развивать территориально разветвленную инфраструктуру программного и технического обслуживания микроЭВМ.

Выводы

1. Более 80 % средств, расходуемых промышленно развитыми странами на производство и эксплуатацию всех типов и классов ПЭВМ, направляются непосред-

ственно в сферу профессиональной человеческой деятельности для повышения производительности труда.

2. Расходы на выпуск «домашних ПЭВМ» и производство пакетов «игровых программ» представляет собой в значительной степени расходы на рекрутирование новых контингентов пользователей профессиональных ПЭВМ.

3. Основные области профессиональных приложений ПЭВМ: первого поколения (8-разрядные) — обучение; второго поколения (16-разрядные) — автоматизация учреждений; третьего (32-разрядные) — САПР, АСУ ТП.

4. Чтобы крупномасштабный выпуск ПЭВМ оказался для экономики страны, в целом, рентабельным, машины должны быть, с точки зрения пользователя, «всегда работоспособны». Иными словами, в производство экономически целесообразно запускать лишь такие модели ПЭВМ, которые превосходят по уровню эксплуатационной надежности существующие типы мини-ЭВМ по крайней мере на порядок, а уровень надежности больших машин (класса, например, ЕС ЭВМ) на два порядка.

5. Основными факторами, в значительной степени предопределяющими экономические итоги государственной программы массовой компьютеризации народного хозяйства, являются в настоящее время темпы развития и эффективность функционирования инфраструктуры технического и программного обслуживания микроЭВМ.

ЛИТЕРАТУРА

1. Лавров С. С. Кому и для чего нужна персональная вычислительная машина?— В кн. «Персональные компьютеры».— Материалы 1-й Всесоюзной конференции «Диалог-82-Микро» (23—25 ноября 1982, г. Пущино).— Пущино: НЦБИ АН СССР, 1983, с. 5—9.

2. Ершов А. П. Персональная ЭВМ — предок млекопитающих в динозавровом мире ВЦКП.— Там же, с. 9—25.

3. Брябрин В. М. Профессиональные персональные компьютеры.— Там же, с. 25—31.

4. Громов Г. Р. Персональные вычисления — новый этап информационной технологии.— Там же, с. 31—70; Микропроцессорные средства и системы, 1984, № 1, с. 37—50.

5. Ершов А. П. Автоматизация работы служащих.— Микропроцессорные средства и системы, 1984, № 2, с. 6—15.

6. Борковский А. Б. Многооконное текстовое взаимодействие с персональной ЭВМ.— Микропроцессорные средства и системы, 1984, № 4, с. 47—50.

7. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации.— М.: Наука, 1984.—237 с.

8. Bender E. IBM offers dual-processor desktop models of Series / 1.— Computer World, 1985, v. 19, N 10, March 11, p. 4.

9. IBM rolls out XT version.— Computer World, 1985, April 22, p. 12.

10. The IBM PC/AT.— Byte, 1984, October, p. 108—114.

11. IBM PC, PC XT PCjr, 3270-PC and PC/XT-370.— Small Business Computers News, 1983, v. X, N 11, p. 4—31.

12. Dembo K. Business software for personal computers.— Data processing, 1984, v. 26, N 5, p. 34—36.

13. Bhatt P. C. P. Electron.— Ind. and Plan., 1983, v. 10, N 7, p. 485—491.

14. Industry Week. 1984, Sept., 17, p. 69, 111.

15. Surving the PC war.— Modern office technology, 1984, Oct., p. 76—89.

16. Cashman M. Window Orientation Becaming Standart Display Mode.— Digital Design, 1984, Jan., p. 93.

17. Wolf R., Delivery P. The quiet invasion of micro-computers.— Electrical World, 1983, July, p. 49—56.

18. IFIP congress series; v. 9, Amsterdam, Elsevier science publishers, 1983.—976 p.

Статья поступила 18 июня 1985 г.

УДК 155.5:681.3

Е. Е. Лысенко

КОМПЬЮТЕРНАЯ ИГРА С ТОЧКИ ЗРЕНИЯ ПСИХОЛОГА[1]

В последние несколько лет в практике использования ЭВМ появилось интересное явление: игры человека с компьютером. Г. Б. Кочетков (см. с. 16) приводит содержательные данные об основных видах имеющихся в США компьютерных игр, формах их реализации, положительных и отрицательных последствиях их широкого распространения. Следует отметить, что советские психологи давно и плодотворно изучают игровую деятельность. Психологическая теория игры, разработанная в нашей стране, связана с именами Л. С. Выготского, С. Л. Рубинштейна, Д. Б. Эльконина [1].

Какое же место занимает игра в жизни человека и какие игровые формы изучает психология? Для дошкольного возраста игра является ведущей деятельностью, так как именно в процессе игры ребенок усваивает значения и способы употребления предметов, а также различные варианты отношений между людьми (так называемые социальные роли: роль матери, дочки, гостя, врача и др.). Но и повзрослев, человек не отказывается полностью от игры. Ведь творческие виды деятельности имеют игровой характер. В процессе творчества, так же как и в игре, человек осуществляет прогнозирующее опережение, перебирает возможности для выбора наилучшего варианта, принимает решение, анализирует результаты и строит новые действия. И процесс общения людей тоже носит игровой характер: люди выполняют определенные роли, их действия подчинены правилам, они моделируют ситуацию за себя и за партнера, порой между ними разыгрываются конфликты. Эти игровые особенности общения активно изучаются психологами и используются с целью игровой психотерапии.

Отсюда видно, что игра в ее различных проявлениях представляет собой неотъемлемую сторону жизни человека во всех возрастах.

На базе ЭВМ создан **новый вид игр,** в процессе которых человек реализует деятельность многопараметрического управления процессами экономического характера, военными действиями, перемещением объектов, поиском объектов. Психологические эксперименты показывают, что начинать игру следует с довольно огрубленной (в соответствии с уровнем играющего) формы, но постепенно по мере обнаружения человеком ее закономерностей усложнять модель введением большего количества переменных и зависимостей.

Очень важной особенностью игры является то, что в ней особым образом соединены закономерные и случайные процессы. Закономерности заложены в правилах игры, но они не дают возможности предсказать все ходы. В процессе игры человек, пользуясь данными ему условиями, **постепенно** открывает для

[1] Эльконин Д. Б. Психология игры. — М.: Педагогика, 1978 г. — 303 с.

себя закономерности того процесса, которым он управляет. В силу того, что играющему не удается с самого начала выявить все закономерности оптимального управления, возникает довольно длительный захватывающий процесс их поиска, в котором проявляются три важные **особенности игровой мотивации:** желание повторить цикл игры, соревновательность и неутилитарность игры, суть которой в том, что основной интерес в игре направлен на процесс игры, а результат служит лишь средством, оценивающим процесс (в отличие от трудовой деятельности, когда практически необходим именно результат).

Резюмируя все то, что было сказано о психологических особенностях игр с ЭВМ, следует лишь подчеркнуть, что мы имеем дело с деятельностью творческой, характеризующейся высоким напряжением интеллекта и эмоций, направленных на преодоление препятствий, психическим продуктом которой является общая пластичность поведения. И поэтому игра с ЭВМ может выступать для человека в качестве средства адаптации, самосовершенствования, удовлетворения важнейших потребностей личности, в частности, потребности в самоутверждении, средства эмоционального самовыражения, эмоциональной разрядки и в других качествах.

Наиболее важные области использования игр с ЭВМ. С помощью специально созданных цельных систем усложняющихся компьютерных игр можно целенаправленно тренировать профессиональные мыслительные и двигательные действия.

Играя, человек может знакомиться с будущей профессиональной деятельностью и анализировать свои реальные возможности в соответствии со специфическими требованиями профессии.

Игры с ЭВМ являются хорошими диагностическими методиками для определения различных индивидуальных особенностей человека, что также может быть использовано в целях профотбора и профориентации.

Большое место должны занять компьютерные игры в сфере обучения и традиционным школьным дисциплинам. Например, на материале игровых задач можно помочь учащимся освоить только что выученный на уроке закон и т. д.

Важной задачей «компьютерной грамотности» является обучение непрофессиональных пользователей основам программирования. Для них знание языков общения с ЭВМ должно быть не самоцелью, а средством, помогающим усовершенствовать свою профессиональную деятельность. С появлением персональных компьютеров люди различных профессий впервые получают в свои руки средство для увеличения эффективности интеллектуального труда, средство, которое должно помочь специалисту самому формализовать свои профессиональные знания и затем использовать их при решении творческих задач [2]. Помочь специалисту приобрести в компьютере реального помощника должна игровая компонента, которая является существенной особенностью персональных компьютеров. Роль игровой компоненты в становлении технологии автоформализации знаний проанализирована в книге Г. Р. Громова [2]. Важно отметить, что игровая компонента реализуется в нескольких формах: это не только набор программ различных игр, но и такое программное обеспечение, которое позволяет таким образом общаться с ЭВМ и в игровой форме учиться программированию. Основная цель игровой компоненты — пробудить интерес непрофессионального пользователя к программированию и создать такие условия, при которых в наиболее доступной форме могут быть усвоены необходимые для практики знания по программированию (т. е. обучение по принципу «работать играя»). [2]

При разработке игр необходимы знания психологии игрока, психологических механизмов игровой деятельности, учет влияния игр на психику разных групп пользователей ЭВМ. По многим из этих направлений в настоящее время на факультете психологии МГУ ведутся специальные исследования. [3]

Статья поступила
16 июля 1985 г.

[2] Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации. — М.: Наука, 1984, с. 96—146.

[3] Лысенко Е. Е. «Психологический анализ игр с ЭВМ». Тезисы конференции «Психологические проблемы создания и использования ЭВМ». — М. МГУ, 1985, с. 85—87.

УДК 681.3.06

Г. Б. Кочетков

КОМПЬЮТЕРНЫЕ ИГРЫ: СВЕТ И ТЕНИ

В последние годы ни одно нововведение в области электроники не вызывало в США таких бурных и ожесточенных дискуссий, как компьютерные игры. Их нередко называют также видео- или электронными играми. И хотя эти названия широко бытуют в печати и средствах массовой информации, представляется, что наиболее правильным является первое, поскольку в основе всех этих игр лежит использование компьютера в качестве противника или посредника при игре двух или более лиц.

Споры вокруг компьютерных игр охватили самые различные слои американского общества. Иногда они несколько угасают, но затем разгораются с удвоенной силой по мере появления новых игр. Родители активно обсуждают вопрос — нужны ли компьютерные игры современным детям. Врачи выражают уверенность в том, что это увлечение пагубно влияет на физическое здоровье и психику детей, а министерство здравоохранения и социального обеспечения США даже сделало по этому поводу официальное заявление. Педагоги в то же время указывают на полезность некоторых развивающих компьютерных игр. Законодатели же многих штатов подняли свои голоса, чтобы запретить компьютерные игры наряду с другими видами деятельности, направленными на развитие «дурных наклонностей». И в ряде штатов это удалось сделать. Например, в пуританском Бостоне компьютерные игры в публичных местах запрещены законом. И мы не видели там ни одной аркады,— так называются в США залы, где установлены игровые автоматы.

Но в других штатах компьютерные аркады процветают. Миллионы детей и подростков проводят в них часы, из которых слагаются дни, недели, месяцы. С напряженным вниманием они следят за видеоэкранами, на которых разворачиваются фантастические действия: огромная акула настигает свою жертву, гигантская горилла Кинг-Конг похищает прекрасную блондинку, коварные межзвездные бандиты нападают на Землю, в таинственных пещерах гномы и тролли стерегут несметные сокровища и т. д. Вооружившись чудесным оружием, чаще всего всеразрушающим лазером, малолетние

* Очерк написан на основе личных впечатлений автора от поездки в США в начале 1985 года в составе делегации АН СССР.

игроки с помощью компьютера переносятся в эти волшебные миры, чтобы вступить в борьбу со сказочными чудовищами. И пока родители, врачи и педагоги спорили между собой, вопрос о компьютерных играх решила сама жизнь: в кратчайший срок, всего за 10 лет, индустрия компьютерных игр выросла до многомиллиардной отрасли. Квотер за квотером*, американцы ежегодно расходуют на компьютерные игры больше, чем на покупку граммофонных пластинок, магнитофонных записей и билетов в кино вместе взятых [1, с. 65]. Только в 1982 г. было продано более 30 млн. кассет и дисков. В среднем на каждого ребенка в США приходится по 7—8 готовых кассет [2, с. 156]. К этому следует добавить, что многие популярные технические журналы постоянно публикуют распечатки команд для различных игр. Электронные игры стали, таким образом, неотъемлемым элементом современной американской культуры, и с этим приходится считаться их противникам и сторонникам.

Немного истории

Электронные игры берут свое начало от «Космической войны», которая была разработана в начале 1962 г. в Массачусетском технологическом институте (МТИ). Основная цель ее создателя Стива Рассела заключалась в том, чтобы продемонстрировать возможности программирования в управлении видеоэкраном. Эта идея возникла в рамках тесного кружка «одержимых программистов» или «хаккеров» (от англ. hacker), члены которого мечтали создать способ демонстрации магической силы компьютера. Возможность подобной демонстрации появилась с установкой в МТИ первой машины PDP-1, в которой имелся видеоэкран. Местом действия нарождающейся игры был выбран космос, поскольку все хаккеры были страстными поклонниками научной фантастики, и компьютер они также рассматривали как первую ласточку машин будущего.

«Космическая война» в первоначальном варианте была просто программой, позволяющей с помощью четырех запрограммированных клавиш клавиатуры управлять изображением ракеты на экране: изменять траекторию ее вправо/влево, ускорять полет и открывать огонь по противнику. Далее С. Рассел прибег к простому трюку: он оставил программу игры на своем рабочем столе. Это означало, что каждый из хаккеров мог внести в нее любые изменения. И игра быстро превратилась в плод коллективного творчества одержимых программистов из МТИ, их любимое детище и способ провождения свободного времени. Очень скоро все свободное машинное время PDP-1 стало использоваться для игры. Она мгновенно распространилась по всем университетским вычислительным центрам [3, с. 44—53].

Первоначально руководство вычислительных центров пыталось бороться с игрой, считая, что она отвлекает специалистов от серьезной работы, и, что гораздо более существенно, ведет к растрате ресурса машинного времени. Но оказалось, что бороться с «космической войной» невозможно. Игра приобрела полулегальный статус, получила романтический ореол, и продолжала завоевывать сторонников. В конце 60-х годов в США даже проводились турниры по «космическим войнам» [3, с. 201—202]. Но высокая стоимость машин и сложность работы на них не позволяли этой игре в те годы стать массовым увлечением. Более 10 лет она оставалась уделом избранных — специалистов по ЭВМ.

Первая коммерчески доступная всем компьютерная игра появилась лишь в 1972 г. Она называлась «Понг» и была стилизованной версией пинг-понга. В ней двое играющих поочередно отбивали мяч с помощью «ракеток», которые изображались в виде двигающихся по краю экрана полосок. «Понг» был изобретен выпускником университета штата Юта Ноланом Бушнелем, проводившим в студенческие годы в вычислительном центре университета многие часы, сражаясь в «космические войны». Чтобы собрать необходимые для обучения деньги, он подрабатывал летом в городском парке, на аттракционах. Здесь он сделал важное для будущего развития компьютерных игр наблюдение — наибольшей популярностью пользовались те из аттракционов, у которых родители могли свободно оставить своих детей, чтобы на время заняться неотложными делами. Он помнил, какой притягательной силой обладала «космическая война», и задумал сделать игру простой и дешевой. Изобретение в начале 70-х годов микропроцессоров позволило Н. Бушнелю собрать специализированный игровой компьютер. Первой программой для него и был «понг». Для производства и реализации игры Н. Бушнель организовал компанию «Атари», название которой заимствовано из игры «го».

Расчет оказался верным. «Понг» пользовался огромной популярностью, и, как оказалось, именно благодаря способности развлекать детей, пока родители заняты своими делами. Но простота «Понга» оказалась и его слабым местом: на рынке появилось множество игр, которые по-существу ничем не отличались от «Понга», но имели другое внешнее оформление, например электронный хоккей, футбол и т. п. Таким образом, «Понг» положил начало целой отрасли электронных игр, и тем самым открыл двери домов рядовых американцев для новой компьютерной технологии.

Несмотря на то, что «Понг» получил огромную популярность, потребовалось еще около десяти лет, чтобы коммерческие компьютерные игры дошли в своем развитии до уровня сложности своей прародительницы — «Космической войны». Для этого нужно было, чтобы микрокомпьютеры стали такими же мощными и быстрыми, как первые миникомпьютеры, чтобы способы программированного управления экраном достигли определенного уровня сложности и т. п. Таким образом, игра — это не просто развлечение, это также способ демонстрации в наиболее доступном и легко обозримом виде некоторых достижений в области вычислительной техники (цветная графика, программирование сложных видов движения на экране, специальные эффекты, например взрывы, наконец, голосовая обратная связь через синтезатор голоса и т. п.). В современных играх компьютер даже комментирует ходы своего партнера.

Все многообразие современных компьютерных игр, получивших распространение в США, может быть достаточно просто классифицировано: 1) игры типа «убей их всех», в которых главный герой должен победить своих врагов, кого бы они не представляли; 2) игры приключенческие, в которых герой как бы проходит по страницам приключенческих повестей и романов; 3) игры стратегические, в которых требуется принимать решения по изменению стратегии поведения в ходе игры.

«Космическая война» относится к играм первого типа. Последователи этого направления в основном повторяют идеи этой игры, делая более сложным сюжет, добавляя различные трюки в программу, создавая новые звуковые и зрительные эффекты. Вплоть до последнего времени подобные игры преобладали на полках магазинов. К ним относится и самая популярная в США игра — «Роботрон». Действие в ней происходит в 2084 г. За сто лет наука и техника сделали в своем развитии значительный прогресс и создали на земле новую цивилизацию разумных машин, которые должны помогать человечеству. Но что-то произошло в механизме воспроизводства роботов и они начали методически уничтожать людей. И вот наконец осталась одна единственная семья, за которой идет охота. Игрок вступает в борьбу как супермен, вооруженный лазерным оружием, и его задача — спасти земную цивилизацию. Не удивительно, что по популярности

* Квотер — монета достоинством в 25 центов, которую необходимо опустить в игральный автомат.

«Роботрон» втрое превосходит другие игры [1, с. 158].

Вторую по распространенности группу составляют так называемые приключенческие игры. Это, по существу, длительный диалог с компьютером, в который заложен некоторый сюжет, и машина ведет вас по нему, задавая различные ситуации. Игрок должен выбирать каждый раз, как ему быть, что делать. Каждое «приключение» начинается с описания места действия, откуда игрок стартует в мир фантазии. Первоначально диалог с ЭВМ велся в форме печатного текста. Затем был изобретен способ представления описаний в виде изображений, хранящихся в памяти ЭВМ, и появилась возможность зрительного восприятия. Одной из наиболее популярных среди детей игр этого типа стали «Подземелья и драконы», где действуют гномы, тролли и другие сказочные герои сказок, а гигантский дракон сторожит несметные богатства. Задача в том, чтобы похитить сокровища и невредимым покинуть подземелье. Но существуют варианты аналогичных игр для взрослых, сценарии которых разработаны, например, в форме детективов, где нужно найти убийцу, проведя целое расследование.

По существу, игры этого типа не так сложны по своим программистским идеям. Но они требуют хорошей проработки сценария, его многообразных вариантов. Поэтому считается, что возможности такого рода игр станут безграничными, когда к разработке их сценариев будут привлечены талантливые драматурги и сценаристы. По существу, приключенческие игры уже дали начало новому виду искусства, отличного от кино и телевидения, где сюжет развивается последовательно и не имеет вариантов. Уже в настоящее время существуют приключенческие игры, в которых описания ситуаций задаются в динамической форме в виде мультфильма, синтезированного компьютером. Например, в одном из них игрок как бы сидит за рулем автомобиля, двигающегося вдоль улицы. Можно остановить автомобиль у любого дома и войти внутрь и т. п. Любое действие играющего как бы оживает на экране компьютера.

Одна из наиболее распространенных игр, относящихся к разряду стратегических — «Пэк-мен». Ее герой — круглое желтое существо по прозвищу Пэк-мен, своеобразный компьютерный вариант колобка, катается по лабиринту и поглощает разбросанные по нему случайным образом таблетки пищи. Каждая таблетка дает ему определенное количество очков. Но одновременно он должен убегать от четырех чертей, которые прячутся в коридорах лабиринта. В обычной ситуации Пэк-мен слабее черта и столкновение с ним означает для него гибель. Четыре таблетки пищи делают Пэк-мена сверхсильным, а черти превращаются на некоторое время в бледные тени. В этот момент Пэк-мен может уничтожить их. Задача игрока — поглощать таблетки пищи и убегать от чертей. Игра веселая и забавная. Хорошие игроки могут продолжать ее часами. Для этого они должны быстро уметь переходить от оборонительной стратегии к наступлению на чертей, и наоборот. Для принятия правильного решения отводятся секунды — игра не может остановиться, решения должны непрерывно следовать одно за другим.

Любителю поразвлечься, таким образом, предоставляется широкий набор компьютерных игр на все вкусы. Но само многообразие игр, заполнивших полки специализированных магазинов по продаже программного обеспечения для персональных компьютеров, не может объяснить причин их популярности. Скорее наоборот — популярность этого вида досуга породила у предпримчивых дельцов стремление разнообразить свой товар. Поэтому вопрос этот требует более внимательного рассмотрения.

Игра как форма социализации компьютерной технологии

Из истории известно, что игры не раз помогали человечеству быстрее осваивать новые технические идеи. Такую же роль выполняют и компьютерные игры.

Как отмечают многие социальные психологи — исследователи проблем компьютеризации — именно игры выполняли в США роль передовых рубежей в распространении компьютерной культуры, именно с ними связано появление в большинстве домов, близлежащих кафе и закусочных первых компьютеров. На игры была направлена первая волна неприязни старшего поколения, и они же вызвали восторженное обожание молодежи. Переоценить роль и значение компьютерных игр в деле ускорения процесса компьютеризации трудно. Известный в США публицист и футуролог Гарри Стайн, например, проводит параллель между электронными играми и тренажерами [4, с. 89]. Последние нужны для того, чтобы уменьшить риск при подготовке специалистов, работа которых связана с повышенной опасностью (водители, пилоты и т. п.), а также снизить затраты на их обучение. Любая же игра — это своеобразный тренажер социальных отношений. В чем же специфика таких компьютерных тренажеров?

Во-первых, они вырабатывают навыки общения с новой техникой. После нескольких часов игр с ЭВМ не только ребенок, но даже и взрослый теряет страх перед непонятной для него техникой. Это обстоятельство сейчас широко используется корпорациями, активно внедряющими автоматизированные системы: прежде чем перейти к использованию ЭВМ в рабочих режимах сотрудникам дают возможность немного (несколько дней и даже недель) поиграться с машиной в шахматы или любые другие игры. Но с помощью игр нельзя ликвидировать «компьютерную неграмотность», т. е. научить людей программированию и пониманию технических аспектов работы ЭВМ. Они лишь создают у сотрудников внутренний стимул, интерес к новой технике и к освоению этих сторон информатики и вычислительной техники.

В этой связи следует отметить, что машины сами быстро эволюционируют в сторону человека; они становятся «дружественными», с ними теперь уже можно «общаться» на языках, близких к естественным. Нужно лишь научиться свободно пользоваться вычислительной техникой и знать возможные границы ее применения. Компьютерные игры как раз и вырабатывают навык «общения» с новой техникой.

Во-вторых, компьютерные игры создают в семье, школе, везде, куда они проникают, особую эмоциональную среду, соответствующую современному этапу развития научно-технической революции. Современные дети растут вместе с компьютерами. К началу следующего века этот пожизненный союз человека и компьютера должен дать свой эффект. Он проявится главным образом в том, что будет полностью снят барьер неприятия новой информационной технологии. Население будет свободно обращаться с ЭВМ и эффективно применять их в любых сферах. «Мы совершенно неверно полагаем, что разработчики компьютеров, программисты и инженеры-электроники составят ядро будущего компьютерного поколения,— пишет в своей книге «Силиконовые боги» Г. Стайн.— На самом деле оно рождается в среде одержимых конструкторов и увлеченных игроков, которые не могут уже представить свое существование вне компьютерной среды».

В-третьих, в разные периоды истории человечества как отношение к играм, так и само определение того, что они собой представляют, существенно отличалось. Но было тем не менее и нечто общее, что оставалось неизменным в веках. Это подход к играм как своеобразному музею истории человечества. Играли лишь в то, что было уже опробовано и испытано. Любая игровая ситуация имела свой реальный прототип.

Компьютерные игры впервые нарушили справедливость этого положения. С их помощью человечество

сделало попытку вырваться из плена традиций и перенестись в играх в будущее, в воображаемые миры. Новые игры — это способ программирования будущего. Электроника позволяет полностью раскрыться творческим способностям программиста. Манипулируя фантастическим миром внутри компьютера, игроки, как дети, так и взрослые, начинают осознавать свои собственные силы в том реальном мире, который находится за пределами экрана. Этот новый тип постижения мира может в корне изменить всю систему образования!

Свобода выбора курса действия в компьютерных играх настолько велика, что игроки перестают осознавать налагаемые на них игрой ограничения. У них возникает естественное желание создать игру практически без ограничений, превратить компьютерную игру в своего рода конструктор. Но если в современном конструкторе из механических деталей и узлов можно собрать разные варианты машин, то в компьютерной игре «юный конструктор» должен получить возможность создавать новые миры, в которых действуют законы, выдуманные им самим. Как отмечал в этой связи известный американский специалист в области информатики Дж. Вейценбаум: «Компьютерный программист — это создатель миров, для которых он — единственный законодатель. Это же, безусловно, справедливо для разработчика игры» [5, с. 115]. Любой завсегдатай компьютерных аркад благодаря этой возможности «побывал» во всех мыслимых мирах прошлого и будущего и это, безусловно, не может не повлиять на его восприятие действительности.

Традиционные игры всегда были связаны с некоторыми объективными законами природы, которые определями естественно-научные границы допустимого, возможного поведения игроков. В компьютерной программе игра ограничивается лишь способностью фантазировать, которой наделен программист. Объект игры в них может нарушать любые запреты, например перемещаться быстрее скорости света, исчезать и появляться вновь, перемещаться во времени и т. п.

Новое поколение компьютерных игр раскрепощает мышление, позволяет размышлять о немыслимом, переместиться в любой мир, как бы сказочен он не был. Неудивительно, что сценарии большинства игр написаны либо по сказкам, либо научно-фантастическим романам и рассказам.

Важной особенностью компьютерных игр, усиливающей их притягательность, является способность запоминать прошлый опыт, самообучаться. В большинстве игр компьютер не повторяет своих ошибок и постепенно усложняет свое поведение по мере того, как его партнер осваивает стратегию победы в данной игре. Более сложные и дорогие автоматы, устанавливаемые в аркадах, оборудованы даже специальными программами, которые синтезируют поздравление сопернику в случае удачного хода и сожаление в случае проигрыша. Часто эти автоматы хранят в своей памяти имена наиболее сильных игроков, так как перед началом каждой игры игрок должен ввести в машину свои инициалы. Если это имя уже есть в памяти, то автомат выбирает более сложную стратегию игры. В США появились местные и национальные знаменитости, чемпионы по отдельным видам компьютерных игр, например по «Пэк-мену». Это становится сильным стимулом, особенно для детей, для которых известность в мире игроков не менее важна, чем профессиональная слава у взрослых.

Очевидно, что компьютерные игры соответствуют веку научно-технической революции. Традиционная игра нацелена на обучение человека некоторым ролям, которые он должен освоить. При относительно невысокой мобильности ролей она удовлетворяла потребностям общества. Но в наше время роли, которые через 10 ... 15 лет ожидают тех, кто сегодня еще сидит за партой, могут значительно отличаться от существующих. Поэтому и функция компьютерных игр может быть иная: дать прогноз возможного развития социальных ролей, показать динамику их становления и т. п., а не привязывать развитие человека к устоявшейся их структуре. Роль игр в истории культуры в целом в настоящее время серьезно изучается, и представляется, что она значительно более существенна, чем считалось. Как отмечает один из специалистов в области компьютерной культуры, профессор социологии Массачусетского технологического института Шэри Тэркл, электронные игры — это начало социализации компьютерной культуры [1, с. 67].

Игры и программирование

Благодаря некоторым свойствам игр этот вид деятельности стал огромным ускорителем в деле распространения идей программирования. Дети, которые первоначально лишь пользовались программами, составленными другими, начали проявлять стремление к самостоятельному программистскому творчеству. Многие из них указывают в социологических опросах, что они хотели бы иметь свой собственный компьютер для игр. Они предпочитают также получать игры в виде распечатки команд, а не готовых кассет или дисков. В этом случае в программу можно вводить любые изменения и юный программист создает, таким образом, свой собственный, неповторимый вариант игры. Наиболее интересные находки-решения становятся достоянием друзей, и игра начинает жить своей собственной жизнью, развиваясь и совершенствуясь, которая во многом повторяет судьбу «космической войны». Но разница при этом огромна. Творцами «космической войны» был узкий круг одержимых программистов-хаккеров. В 80-е годы кружков юных хаккеров, группирующихся вокруг различных версий какой-либо игры, возникло огромное множество.

Новое поколение хаккеров гораздо моложе. Уже в 10 ... 12 лет многие школьники пробуют свои силы в написании собственных игр. Основные кадры программистов фирм, разрабатывающих игры, имеют возраст до 20 лет. Печать США полна сообщениями о школьниках, получающих огромные гонорары за свои разработки игровых программ.

Следует также сказать, что программирование игры стало популярным способом проведения досуга для «серьезных» программистов и способом получить всеобщее признание для начинающих. Рынок игровых программ — самый массовый рынок математического обеспечения. Его размер уже сейчас оценивается миллиардами и продолжает расти. При себестоимости производства одной копии игры в 5 ... 10 долл. цены на рынке могут колебаться от 30 до 250 долл. и выше. Высокая конъюнктура рынка привела к появлению в США в 80-е годы множества мелких фирм, которые производят игры, активно вовлекая в этот процесс школьников старших классов. Это породило ряд социальных конфликтов, возникших на почве того, что сын-школьник начинает получать от своих программ доход, значительно превышающий зарплату отца и т. п.

В основе компьютерной технологии, как известно, лежит понятие алгоритма, близкое понятию правил игры, поскольку последние представляют собой формальную систему поведения, в которой взаимодействуют две или более сторон (ограниченные процедурами и правилами) с целью получения неоднозначного результата [6]. Причем характер взаимодействий игроков определяют не средства осуществления игры, а система правил, следуя которым участники игры «переходят» как бы в воображаемый мир игры. Хорошая игра — это замкнутая система, границы которой строго очерчены этими правилами, и это дает возможность провести много параллелей между игрой и программированием, вычислительной техникой. Труд программиста можно рассматривать как процесс игры по некоторым правилам, и наоборот, игру — как разновидность программист-

ской деятельности. Это обстоятельство проливает новый свет на роль и место компьютерных игр в современных условиях.

Игра рассматривается, таким образом, как начальная ступень на пути в мир новых отношений человека и машины, которые подчинены законам программирования. В ходе игры между игроком и машиной устанавливается сложный комплекс различных связей, не отмечавшийся психологами ранее в отношении других технических систем и нововведений.

Каждая игра — это сложный, активно взаимодействующий с игроком микромир, фантастический, но вместе с тем реально осязаемый на экране. Но все игровые микромиры обладают одним общим свойством: законы, действующие в них, поддаются общим правилам программирования. В результате игры интенсифицируется взаимодействие человека и компьютера, между ними начинается своеобразное соревнование. Как показывают исследования американских социальных психологов, игроки имеют тенденцию одушевлять своих электронных противников. В одном из экспериментов группе студентов была предоставлена возможность играть с ЭВМ без каких-либо ограничений, и все их разговоры записывались на магнитофон. Исследователи обнаружили, что студенты присваивают компьютерам мужские имена и говорят о них в третьем лице, как о живых существах. Вырабатывается особый язык общения с ЭВМ и разговора о них. Во многих чертах он тождествен профессиональному жаргону программистов и других специалистов по ЭВМ.

Поверхностные наблюдатели часто сравнивают игры с телевидением (отсюда и распространенное название — видеоигры). Но это сравнение основано на чисто внешнем сходстве — наличии телеэкрана в обоих случаях. Все остальное, за пределами экрана, у игр и телевизионных приемников различно. И сами игроки никогда не проводят таких параллелей. Они сравнивают компьютерные игры со спортом, медитацией и т. п. Наблюдая телевизионную передачу, зритель остается пленником режиссера. Он может лишь безвольно следовать за ходом событий на экране. В играх же он сам конструирует сценарий. От его воли зависит, настигнет на экране акула свою жертву или нет, схватит ли дракон отважных искателей сокровищ или последние удачно избежат встречи с ним и т. д.

Аналогия игр и телевидения ведет к одному, неверному и, более того, опасному выводу о том, что игры, как и телевидение,— малопродуктивная бездумная растрата времени. Но если в отношении телевидения многие после некоторого раздумья вынуждены будут согласиться с этим утверждением, то для игр оно безусловно не верно. В играх игрок активен, он участвует в конструировании хода событий. Чтобы выиграть, нужны значительные напряжения воли и разума, нужно уметь выбирать победные стратегии в сложной динамической обстановке.

Большинство игр допускает несколько уровней сложности и на более высоких из них нужно не только хорошее знание игры, но и высокую сноровку в управлении компьютером — пальцы игрока должны автоматически двигаться по клавиатуре панели управления. Как пальцы музыканта по инструменту.

В целом, следует, очевидно, сделать вывод о том, что компьютерные игры позволяют ввести детей, студентов, а также и взрослых в мир компьютеров, наиболее быстро и эффективно знакомят их с новой нарождающейся культурой информационного века и даже вырабатывают некоторые стереотипы поведения и роли, соответствующие этой культуре. «Компьютерные игры,— считает, например, психолог из МТИ Ш. Тэркл,— это мост между реальностью и программированием, большой компьютерной культурой» [1, с. 79]. Поэтому было бы разумно поставить на обсуждение вопрос о необходимости развития компьютерных игр как части большой комплексной программы по ликвидации «компьютерной неграмотности». Должна быть создана целая система развивающих игр, которая могла бы стать серьезным подспорьем в деле компьютеризации школы. Наконец, создание такой отрасли позволило бы активизировать творческий потенциал программистов, что также очень важно в современных условиях.

ЛИТЕРАТУРА

1. Turkle Sh. The second self.— N. Y.: Simon and Shuster, 1984.
2. Rochester G., Gantz Y. The naked computer.— N. Y.: William Morrow, 1983.
3. Levi S. Hackers. Heroes of the computer revolution.— Garden City: Anchor Press/Doubleday, 1984.
4. Stine H. The Silicon gods.— N. Y.: Dell, 1984.
5. Weizenbaum Y. Computer power and human reason.— San Francisco: W. N. Freeman, 1976.
6. Импакт. Наука и общество, 1984, № 2, с. 19.

Статья поступила 5 апреля 1985 г.

НА КНИЖНОЙ ПОЛКЕ

Кейслер С. **Проектирование операционных систем для малых ЭВМ:** Пер. с англ. — М.: Мир, 1986 (II кв.). — 39 л.

Рассматриваются методы проектирования операционных систем (ОС) многоуровневого управления малыми вычислительными машинами, предназначенными для использования в качестве персональных компьютеров или для АСУ ТП. Описываются основные блоки ОС, их назначение, взаимодействие ОС с пользователем. Приведены многочисленные примеры.

Нагао М., Катаяма Т., Уэмура С. **Структуры и базы данных:** Пер. с япон. — М.: Мир, 1986 (III кв.).

Излагаются теоретические основы построения структур и методы проектирования баз данных, получившие широкое распространение в Японии, что представляет значительный интерес в свете создания микрокомпьютеров пятого поколения. Рассматриваются аспекты построения структур баз данных абстрактного типа. Выдвигается концепция независимости базы данных от вида примера. Все конкретные примеры даются на языке Паскаль.

Хирн Д., Бейкер М. **Микрокомпьютерная графика:** Пер. с англ. — М.: Мир, 1986, (III кв.).

Излагаются методы машинной графики с применением рабочих программ на языке Бейсик, с помощью которых можно выполнять чертежно-конструкторскую документацию плоских и пространственных изображений.

Кейтер Дж. **Компьютеры — синтезаторы речи:** Пер. с англ. — М.: Мир, 1985 (IV кв.).

В научно-популярной монографии излагаются современные методы создания искусственной речи. Общие принципы построения систем синтеза речи дополняются описанием конкретных схем электронных блоков, используемых в реальных системах.

УДК 681.322.068

А. Е. Корчак

ИНТЕРПРЕТАТОР ЯЗЫКА ПРОГРАММИРОВАНИЯ БЕЙСИК/F ДЛЯ МИКРОЭВМ

Язык программирования Бейсик, разработанный в 1965 г., в настоящее время получил широкое распространение в связи с бурным развитием микропроцессорной техники и персональных компьютеров. В своем развитии Бейсик прошел путь от простейшего языка программирования, предназначенного для обучения азам программирования и решения несложных инженерных задач и состоящего из полутора десятков операторов, до мощного многоцелевого языка, который используется для решения вычислительных и экономических задач, создания информационно-поисковых систем и баз данных, для обработки текстовой и графической информации, автоматизации научных исследований, проектирования и обучения, отладки аппаратных средств. Чтобы решать столь разнообразный круг задач, в язык Бейсик были введены многочисленные расширения, причем отсутствие стандарта привело к существенным отличиям разных версий Бейсика.

На Рижском производственном объединении ВЭФ им. В. И. Ленина разработана версия языка Бейсик под названием Бейсик/F. Этот язык сравнивается с наиболее популярной версией языка Бейсик фирмы Microsoft (МикроБейсик) для ЭВМ на базе микропроцессора Intel 8080 [1].

В 1980—1981 гг. создан ряд версий Бейсика для микроЭВМ «ВЭФОРМИКА» (микропроцессор КР580ИК80), серийно выпускаемой на РПО ВЭФ, которые поддерживали перфоленточный ввод-вывод [2]. В 1981—1983 гг. на базе языков Бейсик фирм Microsoft [1] и WANG [3], а также с учетом предложений по новому ANSI-стандарту Бейсика [4] разработан язык программирования Бейсик/F и реализован интерпретатор в операционной системе DOS/F автоматизированного рабочего места «Экран» (на базе микроЭВМ «ВЭФОРМИКА» с НМД «ИЗОТ-1370») [5].

В 1984 г. разработаны усовершенствованная версия языка Бейсик/F (версия 5) и интерпретаторы для работы в операционных системах CP/M2.2 [6], ОС1800 [7], МикроДОС [8], CP/M Plus3.1 [9], ISIS-II [10], ДОС 1800 [7]. Создана программа настройки интерпретатора на конкретный компьютер, дисплей, принтер.

Интерпретатор языка Бейсик/F можно использовать как для решения простых инженерных задач, так и для создания больших программных систем, чему способствует модульность языка и средства фазирования программ. Для реализации программ, работающих в диалоговом режиме, для которых приемлемо быстродействие интерпретатора, а также для программ, текст которых часто изменяется, использование интерпретатора имеет свои преимущества. Функции редактирования текста, запуск программы и отладка сосредоточены в интерпретаторе, что значительно ускоряет цикл подготовки программы к выполнению по сравнению с использованием компилятора, где требуется работа редактора текста, транслятора, редактора связей и, наконец, самой программы.

Язык программирования Бейсик/F имеет ряд преимуществ по сравнению с другими версиями Бейсика для микроЭВМ (в частности, МикроБейсик [1]):

возможность модульного построения программ, что обеспечивает аппарат внутренних и внешних процедур с локальными переменными;

реализация управляющих структур, позволяющих использовать методы структурного программирования при разработке программ;

развитая файловая система дает дополнительные преимущества при программировании экономических задач;

работа с целыми данными повышенной разрядности (до 2^{2039} по абсолютной величине);

более гибкая работа с символьными данными: задание максимальной длины переменной либо динамическое распределение памяти под значения переменной с использованием оптимизирующих алгоритмов выделения свободной памяти;

редактирование строки программы с помощью управляющих клавиш дисплея;

работа с исходными текстами программ в коде КОИ-8.

В языке Бейсик/F достигается высокая степень совместимости с языком МикроБейсик благодаря преемственности многих черт языка, а также разработке конвертора исходного текста программ, написанных на языке МикроБейсик, в программу на языке Бейсик/F.

Данные. В языке Бейсик/F три типа данных: *вещественные, целые, символьные. Вещественное* значение меняется в пределах $10^{-38}...10^{38}$ и во внутреннем представлении занимает 4 байт. При печати происходит округление до 6 десятичных цифр. *Целые* данные меняются по абсолютной величине от 0 до 2^{2039}. При печати в десятичном виде допускается до 254 десятичных цифр. Целые константы могут задаваться в деся-

```
   BASIC/F INTERPRETER  VERSION 5.10
 10 ! ********************************
 11 ! *  СОРТИРОВКА ЭЛЕМЕНТОВ В ФАЙЛЕ  *
 12 ! *     В ПОРЯДКЕ ВОЗРАСТАНИЯ      *
 13 ! ********************************
 20 DIM MSF(2000)\ !ЧИСЛО ЭЛЕМЕНТОВ <2000
 30 INPUT "ВВЕДИТЕ ИМЯ ФАЙЛА";FILENAME$
 40 ASSEN #1,FILENAME$
 50 OPEN #1
 60 NUM=0\ ! NUM-ЧИСЛО ЭЛЕМЕНТОВ ФАЙЛА
 70 FOR
 80   NUM=NUM+1
 90   READ #1;MSF(NUM)
100 NEXT WHILE EOF(1)=0
110 NUM=NUM-1
120 CLOSE #1
130 CALL SORT_MAS(MSF(),NUM)
140 OPEN #1,TYPE=NEW
150 FOR I=1 TO NUM
160   WRITE #1;MSF(I)
170 NEXT I
180 CLOSE #1
190 ! ------------------------------
200 PROC SORT_MAS(MAS(),N_OF_EL)
210 ! ********************************
211 ! *  УПОРЯДОЧЕНИЕ   ЭЛЕМЕНТОВ   *
212 ! *    МАССИВА ПО ВОЗРАСТАНИЮ    *
213 ! * MAS() - ИМЯ МАССИВА,         *
214 ! * N_OF_EL - ЧИСЛО ЭЛЕМЕНТОВ   *
215 ! ********************************
230 FOR
240   P1=0
250   FOR I=1 TO N_OF_EL
260     IF MAS(I) > MAS(I+1) THEN BEGIN
270       SWAP MAS(I),MAS(I+1)
280       P1=1
290     END IF
300   NEXT I
310   IF P1=0 THEN EXIT FOR
320 NEXT
330 END PROC
```

тичном (53 %), двоичном (1101B) и 16-ричном виде (OFFH). *Символьные* данные представляют собой последовательность от 0 до 255 символов. Символьная константа задается набором символов в кавычках или апострофах («CONSTANT»), последовательностью символов в двоичном («1011»B) или 16-ричном («1F»H) виде.

Имя переменной, массива, нестандартной функции (начинается с FN) содержит до 31 буквы, цифры, знака подчеркивания. Тип имени задается либо явно, специальным знаком после имени (! — вещественный, % — целый, $ — символьный), либо неявно, по первой букве имени, в зависимости от того, какой тип сопоставлен с этой буквой операторами определения типа DCLFLP, DCLINT, DCLSTR. При определении целых и символьных переменных в операторе DIM можно указать максимальную длину, которая выделяется для хранения ее значения. Это экономит память и ускоряет интерпретацию, если длина присваиваемых значений меняется незначительно. Если максимальная длина не указана, то для значения переменной память выделяется динамически от 0 до 255 байтов в зависимости от присваиваемого значения. Этот режим удобен, если длины присваиваемых величин (например, фрагменты текстов) меняются в широких пределах.

По сравнению с интерпретатором МикроБейсик улучшен алгоритм сжатия «кучи», где накапливаются значения переменных. Для сжатия в интерпретаторе МикроБейсик приходится просматривать n^2 раз таблицу имен, где n — число переменных и элементов всех массивов. Поэтому при работе с символьным массивом из 1000 элементов при сжатии наблюдается задержка интерпретации на несколько минут. В интерпретаторе Бейсик/F к значению переменной добавляется обратная ссылка в таблицу имен, и сжатие происходит с помощью одного просмотра «кучи» (а не таблицы имен).

Массивы в языке Бейсик/F — одно- и двумерные. В операторах присваивания, ввода-вывода можно работать со всем массивом как с единым целым, например, 10 A( ) = 0, 20 PPINT MAS(,);. Псевдопеременные, которые формируются с помощью функций выделения подстроки, дают возможность работать не со всем значением символьной переменной, а с его частью, например 10 MID $ (X $, I, J) = '1'.

Процедуры и поддержки. В языке Бейсик/F *аппарат процедур* позволяет реализовать идеи модульного программирования. Программу можно разбить на модули (процедуры) с независимыми локальными переменными, описаниями типов, функций, блоков данных (формируются операторами DATA). Связь между процедурами осуществляется через параметры, передаваемые при вызове процедуры, и глобальные переменные, объединенные в COM-блок.

Программа на Бейсике состоит из главной программы, после которой могут идти одна или несколько процедур, называемых *внутренними*, так как в момент вызова они находятся в оперативной памяти. Процедура начинается оператором PROC. В нем указано имя процедуры и список формальных параметров, состоящий из переменных и массивов. Оканчивается процедура оператором END PROC. Процедура вызывается оператором CALL (см. программу), в котором указано имя процедуры и список фактических параметров. Передача параметров осуществляется «по ссылке», что позволяет вызванной процедуре изменять данные вызвавшего ее модуля. Количество, тип соответствующих параметров должны совпадать.

Во время выполнения программы можно вызвать *внешнюю* процедуру, находящуюся на диске. В момент ее вызова (оператор CALL) программа в памяти заменяется программой, считанной с диска (внешний модуль), которая состоит из одной или нескольких процедур. Управление и параметры передаются первой процедуре, называемой внешней по отношению к вызвавшему ее модулю. При возврате из внешней процедуры в памяти восстанавливается вызвавшая ее программа, выполнение которой продолжается. Из внешней процедуры могут быть вызваны другие внешние процедуры. Имени внешней процедуры в операторе EXTERNAL противопоставляется имя файла, где находится внешний модуль.

Аппарат процедур позволяет независимо создавать различные модули программы, накапливать и использовать библиотеки процедур, повышать надежность программных комплексов. При использовании внешних процедур можно значительно увеличить размеры программ, создаваемых на языке Бейсик/F.

Оператор CALL служит также для вызова подпрограмм, написанных на других языках, имеющих компиляторы (ассемблер, PL/M и др.). В операционных системах, совместимых с CP/M, *аппарат поддержек* реализован следующим образом. В операторе SUPPORT задается имя файла, в котором находится перемещаемый объектный модуль (тип PRL), и таблица символов (тип SYM), полученные программой LINK. В операторе SUPPORT также указывается список имен подпрограмм этой поддержки, которые должны присутствовать в файле символов (типа SYM). При выполнении оператора SUPPORT автоматически выделяется место в памяти, куда загружается объектный модуль из файла типа PRL, а с именами подпрограмм сопоставляются их абсолютные адреса. Оператор CALL вызывает соответствующую подпрограмму по имени и передает ей список параметров. Описанный механизм избавляет программиста от забот по выделению памяти (как в МикроБейсике). В программе на Бейсике не указываются абсолютные адреса подпрограмм, поэтому изменения в этих подпрограммах не приведут к коррекции Бейсик-программы.

Файлы. Язык Бейсик/F имеет развитые средства для обработки данных, находящихся на диске. В программе можно одновременно использовать 10 файлов. Операторы ввода-вывода работают с *логическими* файлами, которые обозначаются номерами от 0 до 9. Сопоставление с логическим файлом конкретного дискового файла осуществляется оператором назначения ASSGN, например 10 ASSGN # I, «FILE1.TX1». Отмена назначения выполняется оператором ASSGN # I, *. После назначения файла его необходимо открыть с помощью оператора OPEN, задающего режимы работы с файлом, который может открываться как потокоориентированный или записеориентированный. *Потокоориентированный* файл содержит данные в символьном представлении, при вводе (выводе) (операторы READ и WRITE) происходит преобразование из символьного представления во внутреннее, и наоборот. Операторы LINE READ и LINE WRITE предназначены для работы с текстовыми файлами.

Записеориентированный файл состоит из записей одинаковой длины, содержащих данные во внутреннем представлении. При чтении (оператор GETR) содержимое записи без преобразований последовательно пересылается в области значений элементов списка ввода. При выводе записи (оператор PUTR) значения элементов списка вывода во внутреннем представлении образуют содержимое записи, которая пересылается в файл. Длина вещественного значения равна 4, а длина символьных и целых данных определяется их значениями либо задается функцией SIZE.

Различают четыре типа файлов: *вводные, выводные* (с дописыванием информации к концу файла), *новые выводные* (с предварительной очисткой файла), *модифицируемые*. Для вводных и модифицируемых файлов оператор RESTORE устанавливает указатель файла в начало, а оператор IGNORE пропускает указанное число записей. Для модифицируемого файла оператор BACKSPACE возвращает указатель файла на предыдущую запись, а оператор ENDFILE усекает файл.

По типу доступа различают *последовательные* и *прямые* файлы. В прямом файле записи пронумерованы, разрешается произвольный порядок работы с записями. Оператор SET

устанавливает указатель файла на заданную запись, а оператор SKIP сдвигает указатель файла вперед или назад. Функция LOC определяет номер текущей записи.

Для удаления и переименования файлов служат операторы KILL и RENAME. Оператор ATTRIB изменяет атрибуты файла. Работа с файлами прямого доступа удобнее, чем в МикроБейсике, где процесс создания буферов (оператор FIELD) и ручное преобразование типов требуют кропотливой работы. К тому же использование символьных переменных в качестве полей буфера может легко привести к ошибкам [1].

Операторы и функции. Язык Бейсик/F имеет средства для программной обработки ошибок и прерываний. Оператор TRAP включает режим программной обработки ошибок и определяет номер строки, с которой начинается подпрограмма обработки ошибок. При возникновении ошибки вызывается эта подпрограмма. Ей доступны номера ошибки и ошибочной строки. После выполнения подпрограммы управление передается ошибочной строке, следующей за ней, или любой указанной строке (оператор RESUME).

Оператор INERRUPT задает действия при возникновении прерывания. Операторы INTERRUPT *e*, ✳ и INTERRUPT ✳ маскируют указанное или все прерывания (*e*-выражение). Если выполнен оператор INTERRUPT e GOTO ⟨метка⟩, то по прерыванию с номером *e* управление будет передано указанной строке. Оператор INTERRUPT *e* GOSUB ⟨метка⟩ определяет следующие действия по прерыванию: выполнение программы приостанавливается, выполняется указанная подпрограмма, после возврата из которой выполнение программы будет продолжено.

Операторы OUT, WAIT, функция INP предназначены для работы с каналами (портами) ввода-вывода [1].

В языке Бейсик/F усилены возможности операторов условного перехода и циклов. В операторе IF после слова THEN (ELSE) можно задавать метку, оператор или блок операторов — последовательность любых операторов, начинающаяся после слова BEGIN и оканчивающаяся операторами ELSE или END IF, например (*e* — любое выражение)

```
IF e THEN BEGIN
⟨последовательность операторов⟩
EISE BEGIN
⟨последовательность операторов⟩
END IF
```

Условный цикл имеет вид (квадратные скобки означают возможное отсутствие параметра)

```
FOR [WHILE e1]
NEXT [WHILE e2]
```

Если $e=0$, то происходит выход из цикла. Для выхода из цикла предназначен оператор

```
EXIT FOR [e],
```

где значение *e* определяет, из скольких вложенных циклов осуществляется выход. При выходе из цикла с помощью оператора GOTO уровень вложения циклов автоматически уменьшается на 1.

В языке Бейсик/F предусмотрена работа с двумя устройствами печати: дисплеем и принтером. Выбор текущего устройства печати и длины строки печати производит оператор (команда) ALST (в МикроБейсике устройство печати фиксируется в операторе PRINT). Оператор PRINT печатает значения числовых и символьных данных и управляет курсором устройства печати. В операторе PRINT можно использовать следующие функции: TAB — установка курсора, SPC — сдвиг курсора влево или вправо, FMT — установка формата печати, HEX, BIN — печать в 16-ричном или двоичном виде, TIM — временная задержка, PAGE — переход в новую страницу, CRT — установка курсора в указанную позицию экрана дисплея. Имеется оператор вывода информации на графическое устройство DISP. Драйвер для конкретного графического устройства пишется пользователем и подключается к интерпретатору.

Для вычисления математических функций, преобразования типов, работы с символьными данными служат 50 стандартных функций. Отметим функции побитового сдвига (влево, вправо), в том числе циклического, чтения символа с консоли, функции сцепления строк, определения текущей даты и времени, текущего положения курсора устройства печати.

Команды интерпретатора. 17 команд интерпретатора управляют его работой. Для удобства оператора имя команды задается введением первой буквы имени. Имеются команды удаления строк, печати листинга, перенумерации строк, печати оглавления диска, текущих режимов работы с файлами. Команда EDIT служит для редактирования строки программы с помощью управляющих клавиш дисплея, обеспечивающих сдвиг курсора влево, вправо, в начало, в конец,

```
IF e THEN BEGIN
⟨последовательность операторов⟩
END IF
```

удаление, вставку символа, удаление до конца строки. Команда GEN автоматически нумерует строки программы при вводе их с консоли. По команде HELP на экран дисплея выводится краткое описание языка Бейсик/F.

Программа на диске хранится в виде исходного текста либо объектного модуля, содержащего текст программы в сжатом виде, его внутреннее представление (польская запись), таблицу имен. Команды INPUT и OUTPUT служат для ввода и вывода программы в указанный файл. По команде INPUT можно ввести текст программы с перенумерацией меток, что облегчает компоновку программы из процедур. При выводе объектного модуля его можно защитить от последующих коррекций и распечатки.

Команда (оператор) TRACE устанавливает режим трассировки, при котором печатаются номера всех выполняющихся операторов. С помощью команд RUN и CONT можно выполнить отдельный участок программы. Выполнение программы можно приостановить нажатием специальной клавиши. С помощью команды LIST D печатаются текущие уровни вложения циклов и подпрограмм и другая информация, полезная при отладке. Интерпретатор имеет режим непосредственного выполнения операторов. Все эти средства облегчают отладку программы.

Настройка. Интерпретатор языка Бейсик/F может работать на любом компьютере с микропроцессором КР580ИК80, U880, Intel 8080/8085, Z80.. Имеется программа настройки, определяющая в диалоговом режиме параметры компьютера, дисплея, принтера. При настройке задаются номера прерываний, которые можно программировать на Бейсике в данной системе, длина строки и число строк экрана дисплея, коды для позиционирования курсора, управляющие клавиши команды EDIT, коды для очистки экрана, действия при выводе служебных символов (коды 00—IF, 7F) и другие параметры. Можно подключить подпрограмму инициализации, позиционирования курсора, драйвер графического устройства. Сообщения, выдаваемые программой настройки и интерпретатором, можно изменить и перевести на любой язык.

Документация поставляется на дискетах и состоит из четырех документов. *Введение в Бейсик* предназначено для начинающих программистов, снабжено примерами, его сведений достаточно для программирования инженерных задач. *Описание языка Бейсик/F* содержит обзор возможностей языка, всех операторов и функций, приведенных в алфавитном порядке. В *Описании интерпретатора языка Бейсик/F* приведены все команды, перечень ошибок и их объяснение. *Руководство по настройке* содержит перечень требований к аппаратным средствам, пояснения по каждому вопросу в диалоге с пользователем.

Реализация. При реализации интерпретатора было решено отказаться от непосредственной интерпретации исходного текста. Сначала выполняется синтаксический анализ исходного текста, и при отсутствии

ошибок создается внутреннее представление исходного текста — массив лексем польской записи [11] и таблица имен, а также отводится память для значений переменных, массивов. Затем начинается интерпретация лексем польской записи. По сравнению с интерпретацией исходного текста (интерпретатор МикроБейсик) на 50...70 % повышается быстродействие. Выигрыш особенно заметен для больших программ. Отделение синтаксического анализа от интерпретации дает прямой доступ к таблице имен вместо последовательного поиска переменной по имени; не происходит каждый раз синтаксического анализа оператора, константы, переменной; переход по метке выполняется без последовательного поиска метки с начала программы. Однако для создания массива лексем требуется память, объем которой составляет 20...50 % объема исходного текста в сжатом виде (ключевое слово заменяется двумя байтами: признак и номер). Объектный модуль создается автоматически по команде RUN. После загрузки объектного модуля в памяти запоминается имя файла, из которого он был считан. Если в момент вызова внешней процедуры объектный модуль в памяти не был нарушен после чтения из файла, то он считывается в память из этого же файла. В этом случае не выполняется запись на диск, что ускоряет вызов внешних процедур.

Интерпретатор состоит из корневой фазы, выполняющей начальную установку параметров интерпретатора, фазы редактирования и синтаксического анализа (первая фаза), фазы интерпретации (вторая фаза) с тремя подфазами, в которых реализованы операторы, выполняющиеся сравнительно редко. Интерпретатор языка Бейсик/F написан на языке ассемблер микропроцессора КР580ИК80, его объем составляет 44 К байт. В памяти интерпретатор занимает 19 К байт. При редактировании текста, выполнении команд в памяти находится первая фаза. После синтаксического анализа (команда RUN) при отсутствии ошибок в память загружается вторая фаза. После выполнения программы (операторы END или STOP) в памяти снова восстанавливается первая фаза.

Наиболее полно возможности языка Бейсик/F реализованы в операционных системах CP/MPlus [9] и МикроДОС [8].

Перспективы. В настоящее время разрабатывается компилятор языка Бейсик/F. В отличие от языка МикроБейсик компилятор языка Бейсик/F поддерживает все возможности языка, реализованные в интерпретаторе. Планируется разработка интерпретаторов и компиляторов в операционных системах для ЭВМ на базе 16-разрядных микропроцессоров К1810ВМ86.

ЛИТЕРАТУРА

1. Microsoft BASIC-80 release 5.0 Reference Manual.— Microsoft, 1979.

2. Корчак А. Е. Интерпретирующая система — ВЭФОРМИКА — Бейсик».— В кн.: Тез. докл. на XIV науч.-техн. конф. молодых специалистов РПО ВЭФ.— Рига, 1983.

3. Система ВАНГ-2200В. Руководство для пользователей.— Рига: Зинатне, 1974.—256 с.

4. Kurtz T. On the way to standard BASIC.— Byte, 1982, v. 7, № 6, p. 182—218.

5. Корчак А. Е. Интерпретатор языка Бейсик/F в операционной системе DOS/F.— В кн.: Тез. докл. на XVI науч.-техн. конф. молодых специалистов РПО ВЭФ.— Рига, 1984.

6 CP/M Features and Facilities Guide.— Digital Research, 1979.

7. МикроЭВМ СМ-1800: Архитектура, программирование, применение / А. В. Гиглавый, Н. Д. Кабанов. Н. Л. Прохоров, А. Н. Шкамарда. — М.: Финансы и статистика, 1984.

8. Переносимая операционная система для микроЭВМ МикроДОС.— В кн.: Информационные услуги МЦНТИ. Сведения о подсистемах МСНТИ.— Проспект 1985.— М.: МЦНТИ, 1984, с. 17.

9. Dahmke M. CP/MPlus.— Byte, 1983, v. 8, № 7, p. 360—384.

10. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.—272 с.

11. Грис Д. Конструирование компиляторов для цифровых вычислительных машин.— М.: Мир. 1975.—

Статья поступила 1 декабря 1984 г.

КАК УЧИТЬ ПРОГРАММИРОВАНИЮ

УДК 681.3.06

И. С. Лосев

РАЗМЫШЛЕНИЯ ОБ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

Рассмотрим три непересекающиеся класса людей: профессиональные программисты, непрофессиональные программисты и потребители ЭВМ (которым представления об ЭВМ, информатике и программировании нужны не больше, чем телезрителю знакомство с устройством телевизора и процедурой съемки телефильма).

Основным продуктом профессиональной деятельности профессиональных программистов являются программы, непрофессиональных программистов — результаты, получаемые с помощью этих программ. (В определенных ситуациях непрофессиональные программисты вынуждены писать программы, но эти программы не есть конечный результат их труда, а только средство получения этого результата.)

Граница между непрофессиональным программистом и потребителем ЭВМ прослеживается не четко. К потребителям ЭВМ относится писатель, пишущий книгу на ЭВМ; делопроизводитель, вносящий архив в ЭВМ; любой человек, использующий терминалы ЭВМ для получения справок, заказов на билеты и т. д.; те, кто посвящает свой досуг играм с ЭВМ, и многие другие.

Что же будет делать профессиональный программист через 15—20 лет? Прежде всего работать над созданием матобеспечения для новой техники. Процесс построения ЭВМ с новыми системами команд и архитектурами будет продолжаться, и они потребуют нового математического обеспечения. Появятся и займут лидирующее положение ЭВМ с высокопараллельной архитектурой, и процесс разработки и выпуска различных многопроцессорных архитектур станет аналогичен процессу появления многочисленных вариантов однопроцессорных архитектур в начале 60-х годов.

Другая задача профессиональных программистов — создание систем для использования непрофессиональными программистами и потребителями ЭВМ (например, систем для автоматического конструирования программ, языков программирования сверхвысокого уровня и т. д.). Сейчас ясно, что необходимая интеллектуальность таких систем приводит к жестким требованиям на эффективность реализующих их программ, иначе с ними будет неудобно работать.

Наконец, профессиональным программистам предстоит создание сверхэффективных программ, рассчитанных на выполнение с помощью наиболее мощных ЭВМ для решения предельных по требованиям к вычислительной технике задач.

На долю непрофессионального программиста останется решать малые и средние задачи, составляющие основную массу задач, требующих вычислительной техни-

ки. Но решать их придется не с нуля, а используя системы, о которых говорилось выше. Тем не менее, иногда непрофессиональному программисту придется писать короткие и простые программы, как бы адаптируя нестандартную ситуацию к имеющимся средствам.

Ответ на вопрос, что будет делать потребитель ЭВМ с ее помощью, потребовал бы целой книги. Изменения, которые ЭВМ внесет в быт, будут существенно больше, чем изменения, внесенные в наш быт телевизором, а отношение потребителя к ЭВМ не должно слишком отличаться от нашего отношения к телевизору.

Важно понять, что профессиональное программирование — сложная деятельность, для обучения которой даже на начальном уровне требуется около двух лет; непрофессиональное программирование — вещь крайне простая, которой можно овладеть за 10—12 ч занятий, а использование ЭВМ потребителем — не требующее почти никаких навыков приятное и полезное занятие.

Цели обучения профессиональных программистов. Профессиональный программист должен уметь эффективно (быстро и, в идеале, безошибочно) создавать эффективные, т. е. выполняющиеся за минимальное время и на аппаратуре минимального объема, программы. В начальный период развития ЭВМ основной упор делался на эффективность программ в ущерб эффективности программирования. После того, как процессорные мощности возросли, и машинное время стало дешевле, усилия сосредоточились на эффективности программирования.

На пути повышения эффективности программирования было создано структурное программирование, основные идеи которого сформулированы в СССР в начале 60-х годов, а затем открыты повторно и дополнены в США [1—3].

Следующим шагом было введение в программирование техники построения программ с одновременным доказательством ее правильности (правила Хоара, инварианты циклов и т. д.). К сожалению, этот шаг существенно сложнее для школьника, чем первый, так как требует владения техникой доказательств и начатками математической логики в объеме, большем, чем возможно для общеобразовательной школы (но вполне приемлемом для специализированных средних школ). Из высказанных предположений о будущей деятельности профессиональных программистов следует, что проблема эффективности программ вскоре снова станет существенной.

По-видимому, ключ к эффективности программ лежит в области большего изучения и адекватного использования структур данных (основные результаты, касающиеся эффективности программирования, связаны с использованием управляющих структур).

Рассмотрим вопросы технических средств, облегчающих процесс программирования. Одной из целей обучения профессиональных программистов должно быть создание ситуации, при которой программист способен эффективно работать без соответствующих инструментальных средств, поскольку значительному количеству программистов в процессе создания новой техники придется обходиться без них. Для этого программист должен знать, во-первых, из чего слагается хорошая инструментальная система, и, во-вторых, как должно выглядеть нулевое приближение к ней, которое можно просто и быстро реализовать с тем, чтобы получить возможность работать в условиях, по крайней мере, сносных.

Преподавание программирования имеет еще одну цель — **научить думать.** Что значат эти слова?

Программирование, как и всякая другая целенаправленная деятельность, сводящаяся к решению последовательности задач, предоставляет обширные возможности учиться думать. Однако в этом качестве оно ничуть не отличается от курсов математики, физики, химии, биологии и т. д.

Слова «учиться думать с помощью программирования» обретают конкретный смысл лишь в том случае, если мы находим методы и приемы, которым естественно обучать в курсе программирования, и которые будут необходимы для любой успешной профессиональной деятельности.

Можно выделить, по крайней мере, три таких метода: формальное доказательство как способ убеждения в правильности программы, решение задач «сверху вниз» и структурно-модульное представление решения, а также опыт отладки.

Первый метод — один из основных в курсе математики (с той лишь разницей, что убеждение в правильности математических утверждений заменяется на убеждение в правильности программ) и вряд ли сегодня этому методу проще обучать в курсе программирования, а не математики. Вместе с тем методология модульного подхода и навыки конструирования «сверху вниз» возникают именно в курсе программирования, будучи при этом неоценимыми в различных областях науки и техники.

Очень важным для развития способности думать является и опыт процесса отладки. Он представляет собой сжатую и спрессованную во времени модель естественнонаучной работы. Действительно, цель отладки — это по поведению системы (неисправной программы) выдвинуть гипотезу о механизме, обеспечивающем такое поведение (неверные команды), и сформулировать эксперимент (тест), проверяющий данную гипотезу. В отличие от естественнонаучной работы, постановка придуманного эксперимента очень легка, и ответ получается сразу. Это делает процесс отладки настолько важным в педагогическом отношении, что, по-видимому, разумно предлагать школьникам такие задачи, как нахождение заранее предусмотренной учителем ошибки в известной программе, внутренней структуры алгоритма по поведению реализующей его программы.

Опыт реальной отладки создает воспитывающий психологический стимул, очень важный для школьника: отлаживающий программу радуется, найдя ошибку в собственной программе,— это означает, что отладка идет успешно. Опыт, приобретаемый в процессе отладки, важен и с практической точки зрения, поскольку «ошибки будут встречаться, даже если основной упор делается на доказательствах правильности. Мудрый программист строит программу с уверенностью, что правильная программа может быть построена и будет построена, если приложить достаточно усилий и внимания, а затем тщательно отлаживает ее, зная, что в ней должны быть ошибки» [4].

Итак, цели обучения профессиональных программистов: научить школьника думать (познакомить его с методами и способами рассуждений, возникающими в программировании, и имеющими универсальное значение); эффективно писать программы; писать эффективные программы; эффективно отлаживать программы; в простейших случаях создавать хорошие инструменты программирования.

Цели обучения непрофессиональных программистов. Их необходимо научить:

а) понимать, что ЭВМ очень полезна для решения задач, даже в областях, не имеющих ничего общего ни с программированием, ни с математикой;

б) писать не слишком сложные программы, владея каким-нибудь языком программирования, причем требования к этому языку более жесткие, чем в случае обучения профессионального программиста;

в) работать с «мирами, реализованными на ЭВМ» (т. е. с системами решения прикладных задач. Примеры таких «миров», наиболее интересные для школьника, будут обсуждаться ниже);

г) использовать методологию «сверху вниз» и структурный подход не только для написания программ, но и для решения проблем, относящихся к другим областям науки и техники;

д) выдвигать гипотезы о внутреннем устройстве системы по ее поведению и проверять эти гипотезы экс-

периментально (в частности, уметь отлаживать свои программы).

Из названных четырех пунктов к собственно программированию относится только второй. В остальных ЭВМ используется как средство обучения думать.

Цели обучения потребителей ЭВМ. В данном случае цель сформулировать столь же просто, сколь сложно реализовать. Необходимо, чтобы школьнику было интересно и полезно взаимодействовать с ЭВМ, причем не для того, чтобы получить хорошую отметку по программированию, а для того, чтобы решать собственные проблемы, в том числе не связанные со школой.

Способы и средства преподавания программирования. Профессиональное программирование не должно преподаваться в общеобразовательной школе, поскольку требует владения техникой математического доказательства на уровне, не доступном для учеников нематематической школы. Для выпуска профессиональных программистов должны быть созданы специальные школы, в которых, помимо программирования, повышенное внимание уделялось бы математике, в особенности математической логике. Выпускники таких школ обладали бы подготовкой, крайне полезной для поступления на факультеты прикладной математики институтов.

Первая половина курса программирования в специальной школе должна включать в себя основные конструкции программирования и методы эффективного программирования («сверху вниз», структурно-модульное, техника написания программ одновременно с доказательством их правильности, т. е. правила Хоара, инварианты цикла и т. д.), а также основные структуры данных.

Большое внимание должно уделяться эффективности программ и ее связи со структурами данных; процессу отладки собственных программ; поиску структур заданных программ, ответственных за их поведение.

Вторая половина курса программирования в специальной школе должна знакомить учащегося с основными элементами операционной среды: операционная система, загрузчики, трансляторы, средства отладки. Желательно, чтобы выходящий из такого училища программист настолько понимал принципы действия, скажем, транслятора, что был в состоянии написать транслятор с простого языка. Для этого необходимы некоторые знания по математическим дисциплинам, обычно не включаемые в курс математики.

Нужны ли для каждого из двух оставшихся классов пользователей ЭВМ (непрофессиональных программистов и потребителей) отдельные программы преподавания? Нет. Необходима одна программа для всех общеобразовательных школ. Надо с самого начала понимать: как и при преподавании любого другого предмета лишь часть школьников освоит этот материал настолько, что сможет (и захочет) использовать его в своей будущей деятельности.

Таким образом, материал должен быть построен так, чтобы цели обучения, сформулированные выше для потребителей ЭВМ, реализовывались практически для всех учащихся, а в том случае, когда прочитанный курс усваивается полностью, реализовывались бы и цели обучения непрофессиональных программистов. Основное средство — это пакеты программ для удовлетворения школьниками их увлечений вне школьной программы. Приведем примеры.

Игровые пакеты. Известно, что в процессе игры новое усваивается очень быстро и прочно. ЭВМ открывает уникальную возможность построения увлекательных игр, содержащих большое количество «учебного материала». Таким образом, появляется надежда резко интенсифицировать и одновременно автоматизировать (ведь игра может рассматриваться как естественное «обучение без учителя») определенные этапы обучения. В качестве начальных можно использовать игры, не имеющие «учебного наполнения», однако задача создания обучающих игр крайне актуальна. Основная сложность состоит в том, что решать ее должны коллективы, состоящие из высококвалифицированных программистов, педагогов и психологов.

Базы данных по увлечениям. Школьные системы должны быть снабжены базами данных с информацией об известных актерах, популярных эстрадных певцах, хоккеистах и других спортсменах, а также, с каталогами названий и текстов популярных песен.

В курс программирования должны входить сведения о работе с такими базами данных, причем школьнику надо разрешить не только получать из них информацию, но и строить собственные версии, модифицируя и внося в них дополнительную информацию. Приведенные примеры могут и должны быть умножены, однако и из них ясно, какие средства могут научить школьника относиться к ЭВМ как к инструменту, которым полезно и приятно пользоваться.

В **основной курс программирования**, конечно, прежде всего должно входить обучение *языку программирования*. Требования к языку программирования для обучения непрофессиональных программистов существенно более жесткие, чем в случае обучения профессиональных программистов.

В таком языке должно существовать подмножество, на освоение которого от хорошего школьника требовалось бы не более 6 ч занятий и которое давало бы возможность писать короткие программы. Существуют работы, показывающие, что язык Бейсик этому условию удовлетворяет.

Язык должен иметь интерпретаторную версию. При тех скоростях трансляции, которые будут доступны для школьных машин, отладка программы, написанной неопытным программистом, становится безумно скучной работой, сводящейся к непрерывному ожиданию результатов трансляции, и может отбить охоту заниматься программированием у кого угодно. Поэтому необходимо, чтобы измененные программы запускались сразу, без этапа трансляции. Это требование столь важно, что ради него приходится идти на жертвы. В частности, приходится допускать, чтобы часть информации о синтаксических ошибках выдавалась не при вводе строк, а при их исполнении. Очень желательно также иметь и трансляторный вариант языка, полностью совместимый с интерпретаторным, а также режим исполнения операторов с пульта — он полезен, чтобы понимать смысл операторов.

Язык должен обладать средствами, обеспечивающими структурное программирование. Бейсик в стандартном варианте этому требованию не удовлетворяет. В качестве кандидатов может рассматриваться либо некоторая модификация Паскаля или Модула-2 при условии, что для нее удастся выбрать то подмножество, о котором говорилось выше, и будет создан и поставлен интерпретаторный вариант языка, либо варианты «структурированных Бейсиков», близкие к Паскалю (такие, как Бейсик-09).

Кроме языка программирования, в основной курс должны входить *структурная методология построения программ, отладка программ,* в том несколько расширенном смысле, который обсуждался выше, *работа с пакетами программ* для преподавания математики, физики, химии, биологии (машинные эксперименты, стереометрия внутри ЭВМ, а также работа с простейшими пакетами автоматизированного решения химических и физических задач). Именно в этих частях курса школьник получит опыт «жизни в мирах ЭВМ». Однако для реализации такой программы необходимо написать большое число соответствующих пакетов.

На пути внедрения компьютерной грамотности видят много препон: недостаток вычислительной техники соответствующего класса, подготовленных учителей, отсутствие хороших языков программирования. Представляется, однако, что главная трудность — малое количество программ для обучения.

Подобные пакеты не могут быть созданы без участия хороших программистов. Поэтому можно с полным основанием повторить некогда сказанные по другому поводу слова: мало когда в истории будущее столь большого количества людей зависело от столь немногих. Каждый программист, всерьез думающий о будущем, должен часть своих сил потратить на создание программ школьной информатики.

ЛИТЕРАТУРА

1. Гутер Р. С., Арлазаров В. Л., Усков С. С. Практика программирования.— М.: Наука, 1965.

2. Dahl O. I., Dijkstra E. W., Hoare C. A. R. Structured Programming.— N. J.: Academik Press, 1972. (Рус. пер.: Дал О., Дейкстра Э., Хоар А. Структурное программирование.— М.: Мир, 1975).

3. Dijkstra E. W. Go to staitment considered harmfool.— Comm. ACM, 1968, 11 : 2.

4. Грис Д. Наука программирования.— М.: Мир, 1984.

Статья поступила 7 мая 1985 г.

УДК 681.3.06-519.68

Д. В. Варсанофьев, А. Г. Кушниренко, Г. В. Лебедев

Е-ПРАКТИКУМ — ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ШКОЛЬНОГО КУРСА ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

С 1985/86 учебного года в старших классах средних школ СССР введен новый предмет «Информатика и вычислительная техника». В учебнике для 9 класса по этому предмету [1] использован простой, алголоподобный, неформальный язык [2].

В статье излагается формализация ядра этого языка и специально разработанная система для ввода, редактирования и выполнения программ, задач и упражнений из учебника. Система достаточно нетрадиционна, основана на принципе непосредственного редактирования информации [3] и получила название Е-практикум (Е — первая буква первой фамилии в списке авторов учебника).

Система была реализована на механико-математическом факультете МГУ на ЭВМ СМ-4 в ОС RSX11M и использовалась там же при подготовке 250 учителей московских школ.

Обсуждение достоинств и недостатков алгоритмического языка учебника 9 класса или самого учебника не входило в цели авторов. Учебник вышел тиражом около 3 млн. экземпляров и послужил для нас отправной точкой.

Описание языка

Язык Е-практикума (или Е-язык) является формализацией языка, использованного в учебнике. В этом языке, в отличие от языка учебника, вместо $4 \cdot a \cdot c$ надо писать $4 * A * C$, вместо $\sqrt{a} - A ** 0.5$ и т. п.

Проиллюстрируем возможности Е-языка на примере двух программ: заимствованной из учебника программы решения квадратного уравнения и программы нахождения первых 60 простых чисел.

```
АЛГ КВУР (ВЕЩ: А, В, С, Х1, Х2; ЛИТ: У)
  АРГ А, В, С
  РЕЗ Х1, Х2, У
НАЧ ВЕЩ:Д
.  Д: = В ** 2 — 4 * А * С
.  ЕСЛИ Д < 0
.  .  ТО У: = "НЕТ РЕШЕНИЯ"
.  .  ИНАЧЕ
.  .     У: = "ЕСТЬ РЕШЕНИЕ"
.  .     Х1: = (— В + Д ** 0.5) / (2 * А)
.  .     Х2: = (— В — Д ** 0.5) / (2 * А)
.  ВСЕ
КОН

АЛГ ПЕРВЫЕ_60_ПРОСТЫХ (ЦЕЛТАБ:П [1:60])
  АРГ
  РЕЗ П
НАЧ ЦЕЛ:N, К, И, Ц; ВЕЩ:В
.  П[1]: = 2; N: = 1; К: = 3
.  НЦ
.  .  ПОКА N < 60
.  .  И: = 2; Ц: = 0; В: = К
.  .  НЦ
.  .  .  ПОКА И < = N & В <> Ц & В > П[И]
.  .  .  В: = К/П[И]
.  .  .  Ц: = В
.  .  .  И: = И + 1
.  .  КЦ
.  .  ЕСЛИ В <> Ц
.  .  .  ТО
.  .  .     N: = N + 1
.  .  .     П[N]: = К
.  .  ВСЕ
.  .  К: = К + 2
.  КЦ
КОН
```

Программа в Е-языке состоит из секций «алгоритмов». Язык Е-практикума содержит переменные трех типов: целые (ЦЕЛ), вещественные (ВЕЩ) и символьные строки (ЛИТ), а также целые (ЦЕЛТАБ) и вещественные (ВЕЩТАБ) одномерные массивы (в терминологии учебника — таблицы). Все переменные должны быть явно описаны, имена

переменных являются произвольными последовательностями из букв (русских и латинских), цифр и знака подчеркивания («_») и должны начинаться с буквы. В языке есть две управляющие конструкции («ЕСЛИ» и цикл «ПОКА») и оператор присваивания.

Арифметические и логические выражения в Е-языке такие же, как и в обычных языках программирования. Арифметические выражения могут содержать имена переменных, элементы таблиц, целые и вещественные константы (вещественные константы записываются с десятичной точкой), арифметические операции «+», «—», «*», «/» и «**» (возведение в степень), а также скобки «(»и«)». В качестве индексов элементов таблиц можно использовать произвольные арифметические выражения. Предполагается, что все арифметические выражения вещественные, в частности 1/2=0,5. При использовании арифметического выражения в качестве индекса и при присваивании значения арифметического выражения целочисленной переменной или элементу целочисленного массива дробная часть отбрасывается, т. е. А [2/3] воспринимается как А [0].

Логические выражения могут содержать арифметические выражения, связанные знаками отношения «==», «<», «>», «<==» (меньше или равно), «>==» (больше или равно), «<>» (не равно), а также логические операции «&» (и), «!» (или), «/\» (не) и скобки «(»и«)». При вычислении логических выражений правый аргумент конъюнкции («&») или дизъюнкции («!») вычисляется только тогда, когда в этом действительно есть необходимость, т. е. в подвыражениях вида A&B, где А ложно, и A!B, где А истинно, В вообще не вычисляется. Это позволяет писать фрагменты типа

```
ВЕЩТАБ: А [0:9]
. . .
ЕСЛИ И >= 0 & И <= 9 & А [И] > 0
. . .
```

будучи уверенным, что при вычислении условия в ЕСЛИ отказ «индекс вне допустимого диапазона» не произойдет ни при каких значениях И.

Заметим, что для основной программы строкам АРГ и РЕЗ разумно придать смысл операторов ввода-вывода, т. е. в начале выполнения запрашивать у человека значения аргументов, а в конце выводить значения результатов. Однако в Е-практикуме это не сделано: строки АРГ и РЕЗ проверяются на корректность, но игнорируются на этапе выполнения. На причинах такого решения мы остановимся подробнее при рассмотрении процесса исполнения алгоритма в Е-практикуме.

Общая схема работы в Е-практикуме

При запуске Е-практикума можно указать имя файла с программой, созданной в предыдущие сеансы работы в Е-практикуме. В этом случае продолжается работа над старой программой. Если указано имя несуществующего файла, то по окончании сеанса файл будет создан и в него запишется новая программа. Если имя файла вообще не указано, то можно нормально работать — результаты сеанса сохранятся в некотором специальном файле.

Методология взаимодействия Е-практикума с человеком базируется на принципе непосредственного редактирования информации [3]. После запуска Е-практикума человек может непосредственно (как именно — описано ниже) создавать и редактировать свою программу. При этом он взаимодействует с редактором-компилятором Е-практикума, который порождает управляющие конструкции, анализирует каждую введенную или измененную строку, в случае синтаксических ошибок немедленно выдает соответствующую диагностику, а в случае отсутствия ошибок порождает фрагмент интерпретируемого кода для данной строки. Поскольку процесс компиляции совмещен с процессом редактирования, а процесс сборки отсутствует вовсе, то программу в любой момент можно начать выполнять. При этом производится непрерывная визуализация процесса исполнения (также описанная ниже). Выполнение программы в любой момент можно прекратить, отредактировать ее и запустить вновь.

Непосредственное редактирование-компиляция Е-программ

В начале работы с новой программой (при запуске Е-практикума был указан несуществующий или пустой файл) на экране терминала появляются строки:

```
*** Е-ПРАКТИКУМ *** ЛВМ, МЕХМАТ МГУ !
АЛГ                                  !
    АРГ                              !
    РЕЗ                              !
НАЧ                                  !
.                                    !
КОН                                  !
```

и курсор устанавливается за словом АЛГ в первой строке. Только зона правее ключевых слов и точек вертикальной разметки доступна для редактирования, все остальное защищено от изменения. Титульная строка, «забор» из восклицательных знаков и зона диагностики правее этого «забора» защищены также от попадания курсора, т. е. в эти зоны даже нельзя переместить курсор. Можно вводить и ре-

дактировать информацию в строках АЛГ, АРГ, РЕЗ, НАЧ (ни удалить эти строки, ни вставить что-либо между ними нельзя), или вставлять, удалять, редактировать строки и конструкции между строками НАЧ и КОН. Набор команд, который при этом доступен человеку, совпадает с командами непосредственного (экранного) редактора текстов «Микромир-85», разработанного на механико-математическом факультете МГУ [4]. Этот набор включает пошаговые и непрерывные перемещения курсора, замену, вставку, удаление, запоминание и вспоминание символов и строк, контекстный поиск, многократное выполнение команд, повторение предыдущей команды, ввод и выполнение макрокоманд и т. п. К этому общему набору добавлены две специальные команды: последовательное нажатие на клавиши СПЕЦ, Ц вызывает вставку конструкции цикла перед строкой с курсором, а последовательность СПЕЦ, Е — вставку конструкции «ЕСЛИ—ТО—ИНАЧЕ—ВСЕ». Кроме того, изменен смысл двух команд: нажатие на клавишу перехода в начало следующей строки ВК переводит курсор в первую редактируемую позицию следующей строки (т. е. за ключевые слова и точки вертикальной разметки), а нажатие на клавишу удаления строки приводит к удалению конструкции, на первой строке которой стоит курсор.

Например, если из начального положения установить курсор между строк НАЧ и КОН и нажать клавиши СПЕЦ, Е, то на экране появится:

```
НАЧ                    !
. ЕСЛИ                 !        НЕТ УСЛОВИЯ
. . ТО                 !
. .                    !
. . ИНАЧЕ              !
. .                    !
. ВСЕ                  !
.                      !
КОН                    !
```

и курсор будет установлен в первую редактируемую позицию за словом ЕСЛИ. В этот момент можно набрать, например, А < В. При нажатии на клавишу ВК (или при любой другой попытке переместить курсор по вертикали) строка будет проанализирована и справа, в поле за восклицательным знаком диагностика «НЕТ УСЛОВИЯ» заменится диагностикой

```
НАЧ                    !
. ЕСЛИ А<В             ! ИМЯ НЕ ОПИСАНО
```

Курсор при этом будет установлен под первой ошибкой в строке, в данном случае — под неописанной переменной А. Далее, можно, например, переместиться в строку НАЧ и ввести там описания «ВЕЩ: А, В». При покидании строки НАЧ вся программа будет перекомпилирована и поле диагностики в строке «ЕСЛИ А<В» станет пустым. Полная перекомпиляция программы производится лишь при изменении одной из первых четырех строк (АЛГ, АРГ, РЕЗ, НАЧ) и происходит мгновенно. В остальных случаях рекомпилируется лишь измененная строка. С внешней точки зрения, т. е. с точки зрения человека, после покидания строки НАЧ просто исчезнет диагностика в строке ЕСЛИ:

```
НАЧ ВЕЩ:А, В           !
. ЕСЛИ А<В             !
```

После этого можно продолжить редактирование программы либо выполнить ее. В последнем случае на экране появится

```
АЛГ                    !    * ВЫПОЛНЕНИЕ *
  АРГ                  !
  РЕЗ                  !
НАЧ ВЕЩ:А, В           !
. ЕСЛИ А<В             !        НЕ ОПРЕДЕЛЕНО
. . ТО                 !
```

т. е. в строке ЕСЛИ появится диагностика, что переменная не определена, курсор будет установлен под имя этой неопределенной переменной (А), после чего можно продолжить процесс ввода и редактирования Е-программы.

Следует особо подчеркнуть важный принцип, выдержанный при организации диалога в Е-практикуме:

в любой момент человек обладает полной свободой действий.

Это позволяет при создании программы идти к корректному состоянию через некорректные, почти не обращая внимания на предупреждения системы, которые к моменту завершения программы незаметно исчезают сами по себе.

При вставке строк и конструкций они всегда появляются с нужным числом защищенных точек вертикальной разметки. Защищены от изменения, кроме того, ключевые слова и структура конструкции в целом. Последнее означает, что в конструкции

```
НЦ
. ПОКА
.
КЦ
```

ни одну из строк НЦ, ПОКА, КЦ нельзя удалить, между строками НЦ и ПОКА ничего нельзя вставить. Аналогично, в конструкции «ЕСЛИ—ТО—ИНАЧЕ—ВСЕ» удаляема только строка ИНАЧЕ. Поскольку конструкцию можно породить или уничтожить лишь целиком, а редактирование не может нарушить корректность конструкции с точки зрения служебных слов, то некоторый уровень правильности программы обеспечивается автоматически, в частности, невозможны ошибки типа «отсутствует конец цикла».

Таким образом, корректность строки в Е-практикуме, с точностью до описаний переменных, не зависит от контекста и определя-

ется только по данной строке. Редактор-компилятор Е-практикума диагностирует следующие виды ошибок:

СИНТ. ОШИБКА	(например, для строки «А; =В»)
КЛЮЧЕВОЕ СЛОВО	(имена не могут совпадать с ключевыми словами)
ИМЯ УЖЕ ОПИСАНО	
ИМЯ НЕ ОПИСАНО	
МНОГО ПЕРЕМЕННЫХ	(реализационное ограничение)
НЕТ УСЛОВИЯ	
РАЗНЫЕ ТИПЫ	(например, для «А: =2», где А — литерная переменная)
СЛОЖНАЯ СТРОКА	(реализационное ограничение)

Непосредственная интерпретация-прокрутка Е-программ

Непосредственный интерпретатор Е-практикума позволяет выполнять программы в двух режимах: пошаговом и непрерывном. Запуск пошагового режима осуществляется нажатием на клавишу ВЫПОЛНИТЬ (в дальнейшем нажатие на любую клавишу приводит к выполнению очередного шага), запуск непрерывного выполнения осуществляется последовательным нажатием на клавиши НЕПРЕРЫВНО, ВЫПОЛНИТЬ. И пошаговое и непрерывное выполнение в любой момент можно приостановить, продолжить или прекратить вовсе, нажав на специальные клавиши.

В начале интерпретации в зоне диагностики строки АЛГ появляется сообщение «*ВЫПОЛНЕНИЕ*».

В процессе выполнения значения, которые принимают условия в строках ЕСЛИ и ПОКА, и значения, которые принимают левые части операторов присваивания, визуализируются в зоне диагностики на экране терминала всякий раз, как только они вычислены. В пошаговом режиме именно визуализация чего бы то ни было в зоне диагностики считается концом очередного шага. В непрерывном режиме визуализация приводит к тому, что в зоне диагностики начинает «крутиться мультфильм», показывающий ход выполнения Е-программы. Если человек не желает смотреть этот «мультфильм», он может заблокировать визуализацию. В этом случае, по завершении процесса интерпретации, разблокировка произойдет автоматически и на экране будет перерисовано конечное состояние. Разблокировку можно произвести и вручную, после чего показ «фильма» продолжится.

Процесс интерпретации завершается при:

нажатии человеком на клавишу «прекратить выполнение»;

достижении строки КОН (в поле диагностики строки КОН при этом выдается сообщение «***КОНЕЦ***»);

достижении строки с ошибкой этапа редактирования-компиляции (в поле диагностики при этом остается сообщение редактора-компилятора);

возникновении ошибки этапа выполнения.

В последнем случае в поле диагностики строки, при выполнении которой возникла ошибка, выдается сообщение:

НЕ ОПРЕДЕЛЕНО при попытке использовать переменную или элемент массива, которым не было присвоено значение;

ПЛОХОЙ ИНДЕКС, если значение индексного выражения вышло за допустимый диапазон;

*** ОТКАЗ *** при арифметических ошибках.

Например, после завершения исполнения приведенного выше алгоритма нахождения первых 60 простых чисел состояние экрана будет следующим:

```
*****  Е-ПРАКТИКУМ  *****  ЛВМ МЕХМАТА МГУ — ИЮНЬ 1985  !
АЛГ ПЕРВЫЕ_60_ПРОСТЫХ_ЧИСЕЛ (ЦЕЛТАБ:П [1:60])          !  * ВЫПОЛНЕНИЕ *
    АРГ                                                  !
    РЕЗ П                                                !
НАЧ ЦЕЛ:ЧИСЛО_ПРОСТЫХ, К, И, Ц; ВЕЩ:В                    !
.   П [1]: = 2; ЧИСЛО_ПРОСТЫХ: = 1; К: = 3               !          3
.   НЦ                                                   !
.   .   ПОКА ЧИСЛО_ПРОСТЫХ < 60                          !  УСЛОВИЕ ЛОЖНО
.   .   И: = 2; С: = 0; В: = К                           !     281.00
.   .   НЦ                                               !
.   .   .   ПОКА И < = ЧИСЛО_ПРОСТЫХ & В <> Ц & В > П[И] !  УСЛОВИЕ ЛОЖНО
.   .   .   В: = К/П [И]                                 !     16.52
.   .   .   Ц: = В                                       !        16
.   .   .   И: = И + 1                                   !         8
.   .   КЦ                                               !
.   .   ЕСЛИ В <> Ц                                      !  УСЛОВИЕ ИСТИННО
.   .   .   ТО                                          !
.   .   .       ЧИСЛО_ПРОСТЫХ: = ЧИСЛО_ПРОСТЫХ + 1       !        60
.   .   .       П [ЧИСЛО_ПРОСТЫХ]: = К                   !       281
.   .   ВСЕ                                              !
.   .   К: = К + 2                                       !       283
.   КЦ                                                   !
КОН                                                      !  *** КОНЕЦ ***
```

Возвращаясь к вопросу об аргументах и результатах, заметим, что, с одной стороны, непрерывная визуализация процесса выполнения делает ненужным вывод на экран терминала результатов, с другой стороны, возможность после любого изменения программы мгновенно начать ее выполнять позволяет задавать аргументы в виде операторов присваивания в начале алгоритма. Задание аргументов операторами присваивания оказывается даже более удобным, чем их ввод *с терминала*, так как позволяет запустить программу, отредактировав значение одного из аргументов, в то время как при вводе аргументов с терминала их каждый раз надо было бы вводить все заново. Однако отсутствие операторов ввода-вывода в Е-языке не позволяет написать программу, которая ведет диалог с человеком или хотя бы показывает ему какие-то результаты без демонстрации текста программы.

Технико-эксплуатационные характеристики

Е-практикум разработан и эксплуатируется в рамках ОС RSX11M, существенно доработанной в ВЦ отделения геофизики геологического факультета МГУ Г. В. Масловым, А. А. Ребеко.

Разделяемый (реентерабельный) код Е-практикума занимает 4К слов. При ограничениях на сложность Е-программы *«не более 32 строк исходного текста»* и *«не более 120 переменных и элементов массивов»* размер рабочей области пользователя составляет 6К слов. Это позволяет на ЭВМ СМ-4 с памятью 124К слов одновременно работать 12 пользователям (с нулевым временем отклика в режиме редактирования-компиляции, т. е. без свопинга).

Скорость интерпретации Е-программ при одновременной работе 12 пользователей на ЭВМ СМ-4 составляет около 200 строк/с, т. е. приведенная выше программа нахождения первых 60 простых чисел при блокированной визуализации выполняется за 50 с.

Е-практикум реализован на базе непосредственного драйвера текста «НДТ-83» [4]. Благодаря этому он является терминально независимым, т. е. один и тот же код Е-практикума одновременно работает со всеми типами терминалов, поддерживаемыми НДТ. Во внешнем интерфейсе НДТ и, следовательно, в Е-практикуме понятие «тип терминала» вообще отсутствует. Заметим, что эксплуатируемый в настоящее время разделяемый код НДТ (объемом 4К слов) поддерживает одновременную работу на 11 типах терминалов: Видеотон-340, Электроника ИЭ15-00-13, модифицированный (улучшенный) терминал Электроника ИЭ15-00-13, DEC VT-52, ВТА 2000-2, ВТА 2000-3, ВТА 2000-15, Видеотон 52106, Видеотон 52130, Видеотон 52130 на параллельном интерфейсе, Мера 7953. Кроме того, в НДТ легко зарегистрировать терминалы новых типов. В частности, разработка Е-практикума велась на терминале VT-52, а эксплуатация — на терминалах Видеотон-340.

Вопросы реализации

Исходные тексты Е-практикума занимают около 3000 строк на языке MACRO-11.

Трудоемкость — примерно 3 человеко-недели. Работа над Е-практикумом была начата 30 мая 1985 года. 6 июня началась эксплуатация Е-практикума. В течение следующей недели были устранены несколько ошибок и улучшен внешний интерфейс.

По нашему мнению, сжатые сроки разработки Е-практикума объясняются следующим:

у нас был опыт проектирования и разработки подобных систем (α-, β-, γ- практикумы для поддержки начального курса программирования на механико-математическом факультете МГУ). В момент начала разработки мы в деталях представляли, как должны выглядеть редактирование-компиляция и, в общих чертах, интерпретация-прокрутка Е-программ. Лишь визуализация процесса исполнения в Е-практикуме была придумана «на ходу» и отличалась от проектных решений «греческих» практикумов;

Е-практикум реализовывался не с «нуля», а на базе непосредственного драйвера текста «НДТ-83» — базовой системы организации непосредственного алфавитно-цифрового диалога [4]. По нашим оценкам, разработка терминально-независимого Е-практикума с теми же возможностями и технико-эксплуатационными характеристиками без использования НДТ-83 заняла бы около года;

основной единицей реализации был *исполнитель* (пакет подпрограмм в смысле языка Ада [5] или информационно-прочный модуль по Майерсу [6]), а не подпрограмма. Если использование подпрограмм позволяет структурировать множество действий в системе, то использование исполнителей позволяет структурировать все множество объектов системы и операции (действия) над ними. На наш взгляд, выигрыш от использования исполнителей существенней, чем от использования подпрограмм, по сравнению с написанием системы одним куском;

наконец, мы работали в достаточно простой, но удобной инструментальной обстановке. Сюда прежде всего относится система «Микромир-85» — симбиоз непосредственного редактора текстов и непосредственного файлового

монитора с простейшими инструментальными возможностями [4]. Кроме того, использовался разработанный ранее с участием В. В. Бехтерева пакет макроопределений, отражающий принятый на механико-математическом факультете МГУ стиль программирования. Этот пакет позволяет без использования меток программировать управляющие конструкции «если—то—иначе—конец если», «выбор—конец выбора», «цикл—конец цикла», «утверждение» и «отказ». Внутри цикла допустимы итераторы «К раз», «пока» и конструкция «выход». Кроме того, пакет поддерживает структуры «исполнитель» и «предписание», обеспечивает вызов предписаний (подпрограмм и функций) с параметрами, контроль состояния стека и трассировку вызовов при возникновении ошибки.

Итоги эксплуатации Е-практикума

Цель работы в Е-практикуме — изучение основных понятий информатики и вычислительной техники: алгоритм, формальная запись алгоритма (Е-язык), процесс исполнения алгоритма, использование ЭВМ при работе над текстом алгоритма и для его исполнения. Сам Е-практикум при этом изучении служит всего лишь *инструментом* и играет второстепенную, подчиненную роль. Удобный инструмент должен быстро осваиваться и не требовать внимания к себе в процессе работы.

Опыт эксплуатации Е-практикума показал, что, в основном, он является удобным инструментом: после 4 ч знакомства с клавиатурой и редактором текстов новичок успевает за 10 терминальных часов работы в Е-практикуме решить 20—30 упражнений из учебника или задач по мотивам учебника. При этом в процессе работы основные усилия тратятся на алгоритмизацию и исправление логических ошибок, а не на синтаксические проблемы или попытки понять, что же делает написанная программа.

Мы считаем, что работа в такого рода системе по производительности и комфортабельности на порядок превосходит работу в стандартной ОС на Фортране или работу в Бейсик-системе.

В заключение заметим, что столь же производительная и комфортабельная обстановка должна, на наш взгляд, быть создана и для программистов-профессионалов. С этой целью на механико-математическом факультете МГУ в настоящее время ведется разработка учебно-производственной среды программирования «СИМПЛЕКС» [7].

ЛИТЕРАТУРА

1. Ершов А. П. и др. Основы информатики и вычислительной техники. Проб. учеб. пособие для 9 кл. сред. шк.— М.: Просвещение, 1985.
2. Ершов А. П. Алгоритмический язык в школьном курсе основ информатики и вычислительной техники.— Микропроцессорные средства и системы, 1985, № 2, с. 48—51.
3. Лебедев Г. В. Непосредственное редактирование алфавитно-цифровой и графической информации.— М.: ВИНИТИ АН СССР, № 7325—84 деп.
4. Кушниренко А. Г., Варсанофьев Д. В., Дымченко А. Г., Лебедев Г. В. Практическое программирование. Проектирование и разработка диалоговых систем. Нетрадиционный подход.— М.: Изд-во МГУ, 1985.
5. Вегнер П. Программирование на языке Ада.— М.: Мир, 1983.
6. Майерс Г. Надежность программного обеспечения.— М.: Мир, 1980.
7. Кушниренко А. Г. Учебно-производственная система программирования «СИМПЛЕКС». 1. Цели разработки.—М.: ВИНИТИ АН СССР, № 7326—84 деп.

Статья поступила 20 июля 1985 г.

НА КНИЖНОЙ ПОЛКЕ

Липаев В. В. Тестирование программ. — М.: Радио и связь, 1986 (III кв.)—20 л., ил.—В пер.: 1 р. 40 к. 10 000 экз.

Тестирование является основным современным методом достижения высокой корректности и надежности программ, в том числе функционирующих на микроЭВМ. В монографии систематически изложены способы и средства автоматизации тестирования программ на различных этапах их разработки и сопровождения. Обобщены основные теоретические и прикладные работы в области тестирования, опубликованные в последнее время. Значительная часть книги является оригинальной, базируется на исследованиях автора и его учеников.

Проведен анализ сложных комплексов программ как объектов тестирования. Представлены основные статистические характеристики выявляемых в программах ошибок. Введены понятия систематического тестирования программных изделий и упорядоченные категории тестов. Значительное внимание уделено формализации критериев качества, а также оценке сложности и трудоемкости тестирования программ.

Наиболее подробно рассмотрено тестирование программных модулей, что особенно важно для микроЭВМ. Описаны методы и оценка эффективности тестирования модулей без исполнения программы на уровне исходных текстов, тестирования структурных модулей и потоков данных. Для всех методов представлены схемы средств автоматизации тестирования и примеры их практической реализации.

Изложены методы тестирования сложных комплексов программ при отладке и испытаниях в реальном масштабе времени в роботах, ГАПах, при управлении объектами и технологическими процессами. Значительное внимание уделено генерированию тестов, имитирующих внешнюю среду, и обработке результатов динамической комплексной отладки. Отдельная глава посвящена специфике тестирования при сопровождении программ, разработанных как изделия производственно-технического назначения. Даются практические рекомендации по применению основных методов и средств автоматизации тестирования программ.

Для инженерно-технических работников, занимающихся проектированием и эксплуатацией программного обеспечения различных типов ЭВМ, а также студентов вузов и аспирантов соответствующих специальностей.

КОНСУЛЬТАЦИОННЫЙ ПУНКТ "МП"

Уважаемая редакция!

Создавая свою персональную систему на МПК БИС серии КР580, я столкнулся с некоторыми вопросами, на которые прошу помочь мне найти ответ. В системе предполагается применить контроллер ЭЛТ—КР580ВГ75. Поскольку система будет работать «в реальном времени», то данный кристалл будет затрачивать слишком много времени для загрузки своих «буферов ряда» через контроллер ПДП, и при этом блокировать работу процессора. Предлагаю подключить «буферную» внешнюю память объемом до 64 Кбайт, управляемую вторым «внешним» контроллером ПДП, подключенным к одному из каналов контроллера ПДП системы. Обмен данными между контроллерами производится через общий буферный регистр данных в режиме «чтение-запись ВУ». Процессор может обрабатывать сигналы прерывания после окончания ПДП от «системного» и «внешнего» контроллеров... Буферное ОЗУ можно расширить до 256 Кбайт, подключая по сегменту до 64 Кбайт на каждый из каналов внешнего контроллера ПДП. Такое решение пригодно для мультипроцессорных систем и систем обработки и передачи потоков данных. Хотелось бы знать, существуют ли системы с подобной «буферной памятью» у нас или за рубежом? Привожу функциональную схему узлов системы без внешних устройств.

В. А. Руднев
(Кемерово)

Уважаемый Анатолий Викторович!

Схема предлагаемой Вами персональной ЭВМ построена вполне грамотно, и основные идеи, заложенные в ней, работоспособны.

Предложение применить внешнюю (по отношению к локальной памяти процессора) буферную память, используемую другим активным пользователем (процессором, видеоконтроллером и т. д.), широко применяется в мультимикропроцессорных, мультимикромашинных системах и ЭВМ высокой производительности. В частности, это позволило обеспечить эффективную работу (с минимумом взаимных помех) мультимикромашинной системы (12 микроЭВМ «Электроника 60»), рассчитанной на организацию параллельной обработки больших потоков данных.

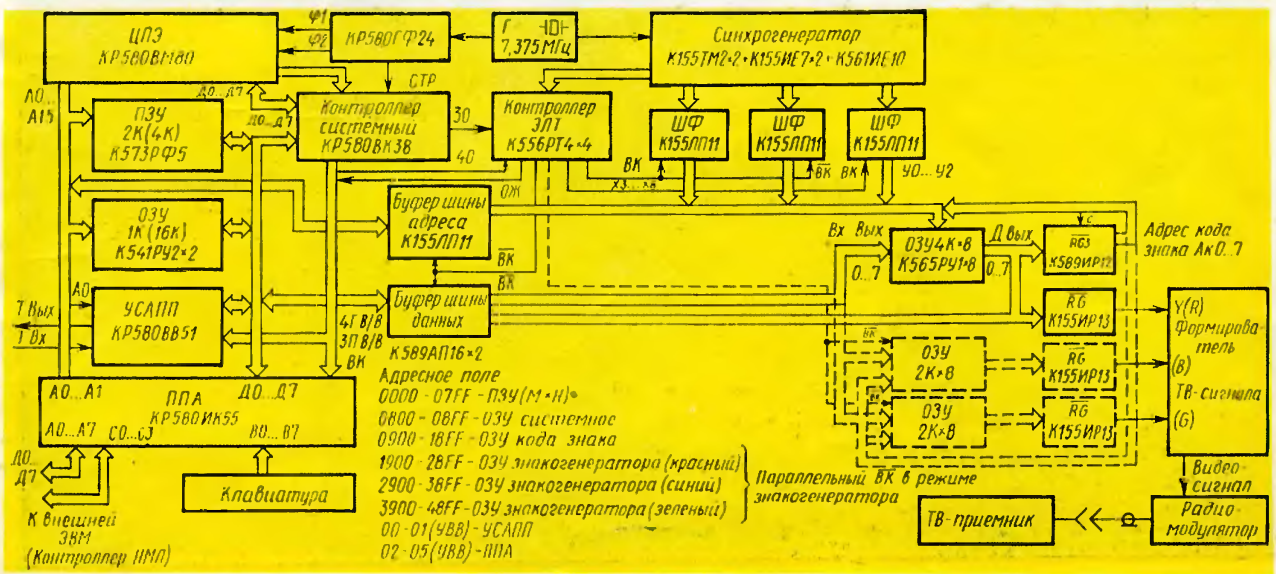
Организация перекачки данных между локальной памятью процессора и буферным ОЗУ с помощью каналов прямого доступа, которую Вы предлагаете, — весьма логичное решение проблемы, хотя и не единственное.

Однако хочется напомнить, что во всякой сложной системе приходится учитывать **реальный выигрыш качества системы** (например, производительности) от введения каких-то дополнительных аппаратных узлов, увеличивающих затраты материальных ресурсов и времени на создание и отладку всей системы. Убедиться в целесообразности того или иного решения можно, как правило, лишь оценив функционирование аппаратно-программного комплекса в целом на примере какой-либо практической задачи. Иначе говоря, решения, связанные с эффективностью работы аппаратуры, должны быть подкреплены эффективностью использования аппаратуры программным обеспечением. Иначе никакого реального выигрыша в эффективности может не получиться.

При создании оригинальной микроЭВМ надо также иметь в виду следующее: на одном из этапов создания аппаратной части микроЭВМ необходимо обеспечить возможность пользования языками высокого уровня, операционными системами и т. п.

Все трансляторы или компиляторы и операционные системы **сильно** ориентированы на какую-либо конкретную структуру микроЭВМ, на особенности ее работы и т. п. Поэтому, создав оригинальную разработку, даже очень хорошую, но несовместимую с уже созданной микроЭВМ, имеющей стандартное программное обеспечение, Вы тем самым возлагаете на себя огромный труд по адаптации существующего или разработке нового системного и прикладного программного обеспечения. Таким образом, на первых порах, мне кажется, целесообразнее попытаться **функционально** копировать какую-либо из серийных микроЭВМ, для которой можно относительно просто использовать программное обеспечение с приемлемой версией ОС.

Ю. И. Торгов
зав. отделом Вычислительного центра
АН СССР



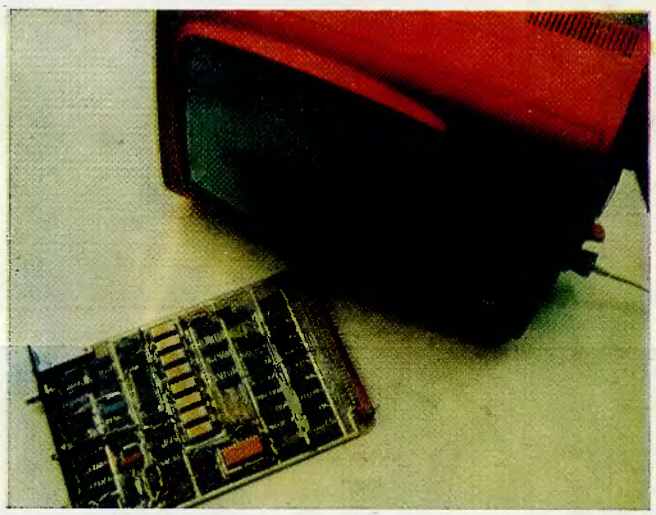
НА ВДНХ СССР

В ПАВИЛЬОНЕ «НАРОДНОЕ ОБРАЗОВАНИЕ»





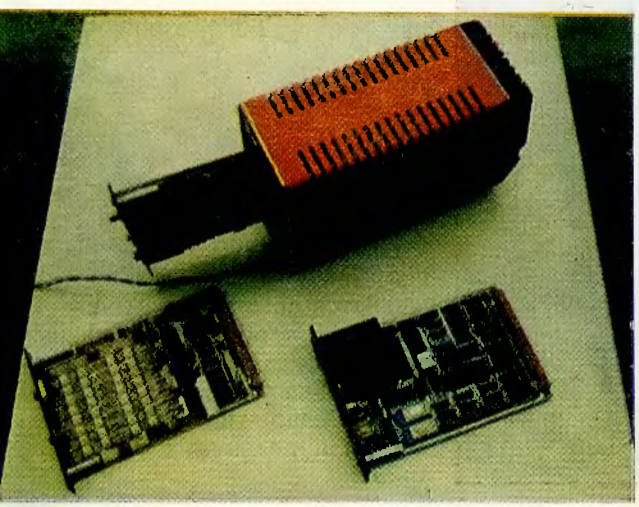
Школьная персональная ЭВМ «Ириша» вызвала большой интерес посетителей выставки



Модули системного блока ПЭВМ

Модуль телевизионного адаптера ПЭВМ

Источник питания системного блока





robotron



Электронная фактурно-бухгалтерская машина Роботрон 1715

С помощью этой машины можно решать задачи фактурирования и бухгалтерской записи. Программное обеспечение позволяет адаптировать ЭВМ к конкретным потребностям различных пользователей.

На лейпцигской весенней ярмарке 1984 г. эта машина награждена золотой медалью.



Терминал бронирования мест К 8927

является комплексным устройством телеобработки данных для решения задач бронирования. Благодаря модульному построению, терминал применяется в различных оформлениях.

УДК 621.317:681.3.62;621.3.049

В. В. Сумин. А. Б. Кислицын, В. И. Воробьев, А. Г. Савин

ПРОСТЕЙШИЕ ПРОГРАММИРУЕМЫЕ ИЗМЕРИТЕЛЬНЫЕ ПРИБОРЫ НА БАЗЕ МИКРОЭВМ

Одна из основных тенденций развития современной измерительной техники — оснащение традиционных средств измерений системами цифрового отсчета и обработки результатов измерений. Дополнение измерительных датчиков цифровыми средствами позволяет повысить точность и надежность измерений, улучшить условия труда, обеспечить оперативную обработку и фиксацию результатов измерений.

Конструктивно устройство цифрового отсчета и обработки (УЦОО) целесообразно выполнять в виде автономного блока, свободно подключаемого к измерительному датчику. Это объясняется двумя факторами:

развитые цифровые средства сложно реализовать в минимальном объеме, требуемом для встраиваемых в датчик устройств;

УЦОО следует ориентировать на работу с различными сменными датчиками ввиду их относительно высокой стоимости.

Построение УЦОО на базе микропроцессорной техники придает измерительным приборам свойство программируемости. В соответствии с формой реализации данного свойства можно выделить приборы, программируемые на уровне входных данных (простейшие программируемые приборы), набора программ (многофункциональные программируемые приборы) и непосредственного программирования (свободно программируемые приборы).

Для неквалифицированного пользователя наиболее удобными являются простейшие программируемые приборы, в которых программируемость ограничивается возможностью ввода и корректировки данных для работы прибора (уставок, ограничений, нормировочных коэффициентов и т. д.). Если ранее программируемые приборы разрабатывались для решения сложных научно-прикладных задач, то настоящее состояние микропроцессорных средств допускает широкое использование простейших программируемых измерительных приборов в производственных условиях.

Рассмотрим принципы организации и реализации приборов данного типа, ориентированных на решение простых, но широко распространенных производственных задач.

Базовый вариант простейших программируемых измерительных приборов

Простейшие программируемые приборы должны обладать малыми габаритными размерами и низкой стоимостью. Для различных вариантов приборов это достигается за счет простоты решаемой задачи, не предъявляющей высоких требований к вычислительной мощности УЦОО, и ее распространенности, обеспечивающей серийный выпуск приборов. Каждый простейший программируемый прибор является универсальным в рамках реализованных в нем алгоритмов измерений, что обусловливается заданием входных данных.

При проектировании конкретных приборов на основе микропроцессорных устройств можно создать универсальную аппаратно-программную структуру УЦОО. Для этого микропроцессорное устройство, подключаемое к датчику для обработки измерительной информации и программируемое впоследствии на уровне входных данных, должно обладать универсальностью своих основных компонентов: аппаратной структуры УЦОО; ввода-вывода и представления исходных данных и фиксируемых результатов измерений; структуры программного обеспечения. Микропроцессорное устройство, обладающее такими свойствами, представляет собой *базовый вариант* УЦОО простейших программируемых измерительных приборов, настраиваемый на конкретный вид прибора добавлением требуемой функциональной программы.

Аппаратная реализация

В соответствии с указанными требованиями в Кировском политехническом институте разработано устройство обработки измерительной

Майоров С. А., Гальперин М. П. Вопросы применения микроЭВМ в приборостроении.— Изв. вузов, сер. Приборостроение, 1983, № 9, с. 72—76.

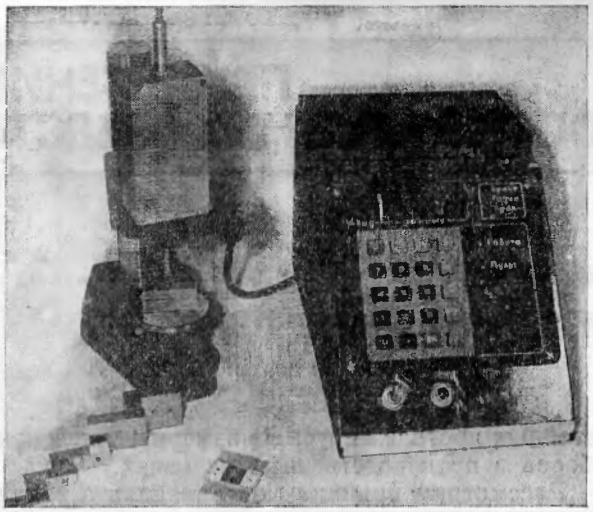


Рис. 1. Базовое устройство обработки измерительной информации

информации (рис. 1) на основе микросхемы К586ВЕ1 (однокристальная микроЭВМ «Электроника С5-31»), которая содержит в себе: ОЗУ емкостью 128 16-разрядных слов для хранения проблемных данных и промежуточных результатов; ПЗУ емкостью 1К слов для размещения несменяемых программ. Для повышения универсальности структуры в устройство введено внешнее по отношению к микроЭВМ сменное ПЗУ емкостью 1К слов.

Обмен вычислительной части с внешней средой осуществляется с помощью следующих схем: начального пуска для запуска микросхемы К586ВЕ1 с лицевой панели; сопряжения с датчиком для обеспечения поступления измерительной информации в микроЭВМ; последовательного интерфейса информации на внешние устройства; управления индикацией и опроса клавиатуры для подключения их к микроЭВМ. Устройство можно настраивать на разрабатываемый вариант прибора путем установки сменного ПЗУ с требуемой программой и при необходимости сменой схемы сопряжения с датчиком.

Организация обмена и хранения данных

Необходимое условие универсальности базового варианта УЦОО — унификация процедуры обмена устройства с пользователем и способа хранения обмениваемых данных. Это облегчает работу с различными видами приборов.

Для эксплуатации устройства приняты два режима работы: пультовой и рабочий. В *пультовом режиме* пользователь осуществляет ввод исходных данных для работы прибора, контроль введенных данных, чтение обобщенных результатов ранее выполненных измерений. В *рабочем режиме* прибор используется для выполненных измерений с индикацией текущих результатов. Все вводимые данные составляют набор входных параметров, обобщенные результаты измерений — набор выходных параметров УЦОО. Входные и выходные параметры размещаются в зоне старших адресов ОЗУ микроЭВМ, причем первыми помещаются входные параметры, что облегчает задание исходных данных. Доступ к отдельным параметрам осуществляется по его номеру в общем списке параметров, последовательность которых и их содержание определяются при реализации конкретного прибора. Для контроля правильности обращения к области параметров в ПЗУ хранятся значения общего числа параметров и числа входных параметров.

Для осуществления обмена с пользователем служит 20-клавишная клавиатура и система индикации, состоящая из семи 7-сегментных светодиодных индикаторов и восьми светоизлучающих диодов. Индикаторная строка разделена на малое 2-разрядное и большое 5-разрядное поля индикации. Пять светодиодов предназначены для индикации состояния прибора, оставшиеся три — для индикации результатов измерений (типовое применение — индикация «Брак +», «Годен», «Брак —»). Клавиатура устройства разделена на фиксированные поля (рис. 2).

Ввод-вывод осуществляется следующим образом. В рабочем режиме по нажатию клавиши *П* происходит обращение к первому параметру УЦОО, в пультовом режиме — к следующему по номеру параметру. Доступ к произвольному параметру по его номеру осуществляется с помощью клавиши *ПР*. При обращении к параметру его номер и значение отражаются соответственно в малом и большом полях индикации. Новое значение входного параметра задается на цифровой клавиатуре с отображением набора по методу бегущей строки с вытеснением старших знаков. Сброс



Рис. 2. Клавиатура УЦОО:

НУ — начальная установка; *П* — пульт; *ПР* — параметр; *О* — очистка; *Р* — работа; *Ф1*, *Ф2* — функциональные клавиши; *ВВ* — внешний вывод

ошибочных значений выполняется клавишей *О*.

При нажатии клавиши *Р* прибор переходит в рабочий режим, в ходе которого могут использоваться клавиши *Ф1*, *Ф2*, *НУ*. Клавиша *ВВ* служит для разрешения вывода информации на внешние устройства в темпе ее поступления.

Указанные обмен и хранение данных приемлемы для самых различных вариантов приборов. Это позволяет выделить неизменную системную часть программного обеспечения.

Программное обеспечение

Для конкретных приборов должна обеспечиваться простота разработки программного обеспечения (рис. 3). Системная (СПНП) и функциональная (ФПНП) программы начального пуска составляют подсистему, обеспечивающую при запуске микроЭВМ исходную подготовку УЦОО к работе. СПНП организует начальную установку схемы УЦОО и системных внутренних переменных. ФПНП устанавливает исходные значения переменных, связанных с решаемой прибором функциональной задачей.

В программное обеспечение УЦОО кроме программ пультового и рабочего режимов входят также подсистемы входного контроля и внешнего вывода, подпрограммы преобразований и вычислений. Последние реализуют в большинстве случаев типовые функции и могут быть разработаны заранее в базовом варианте программного обеспечения. В дальнейшем набор типовых функций можно дополнить специфическими подпрограммами, используемыми подсистемой рабочего режима. Подсистема внешнего вывода включает набор средств, обеспечивающих передачу информации на внешние устройства по принятым в УЦОО форматам данных и сообщений. Аналогично предыдущему эти процедуры могут быть типовыми, разрабатываемыми в составе базового варианта, и специфическими, вводимыми при окончательном формировании программного обеспечения УЦОО конкретного прибора. Подсистема входного контроля содержит набор программ контроля, используемых для проверки исходных данных, вводимых в пультовом режиме. Выполняемый контроль основывается на смысловых проверках и контрольных соотношениях между данными. Жесткая зависимость процедур контроля от исходных данных делает необходимым разработку специфической подсистемы контроля для каждого типа прибора. Независимость размещения программ контроля УЦОО обеспечивается связью их с подсистемой пультового режима через таблицу адресов входов, размещаемую в фиксированном поле ПЗУ.

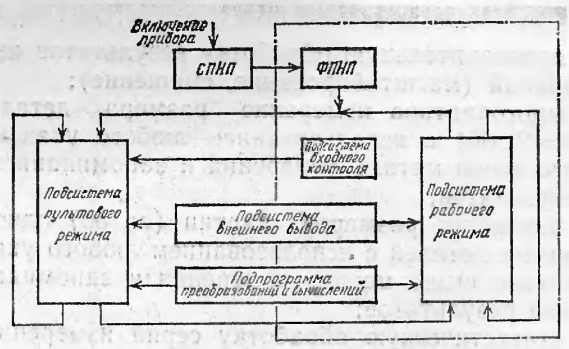


Рис. 3. Структура программного обеспечения УЦОО

Представленная структура отражает стремление к максимально полному разделению программного обеспечения на системную и функционально-зависимую части. Получение нового вида прибора в этом случае связано с разработкой лишь функционально-зависимой части, выделенной на рис. 3 штриховой линией.

Пример реализации простейшего программируемого измерительного прибора

Одним из приборов, реализованных на основе базового варианта УЦОО, является многоточечный измеритель со статистической обработкой серий измерений. Прибор предназначен для работы с датчиками фоторастрового типа с различным шагом дискретности растра, присоединяемыми к УЦОО гибким кабелем длиной до 2 м (см. рис. 1).

Выходные данные прибора: начальная точка измерения (база измерений), шаг дискретности датчика, масштабный коэффициент, коэффициент формы, верхнее и нижнее допустимые значения размеров, допустимое отклонение формы, длина серии измерений. Данные вводятся перед выполнением измерений в пультовом режиме как входные параметры прибора с номерами 00...07.

Последовательные измерения контролируемого размера (размеров) выполняются в рабочем режиме прибора. При этом запоминание результата измерений и переход к очередному замеру осуществляется клавишей *Ф1* (Замер). Функциональная программа прибора позволяет выполнять следующие операции:

абсолютное измерение размеров, в том числе размеров, превышающих рабочий диапазон датчика;

измерение разностей размеров;

измерение отклонений размеров от номинальных значений, задаваемых начальной установкой или ненулевой базой измерений;

настройку на различную дискретность растра подключаемых датчиков;

предварительную обработку результатов измерений (масштабирование, смещение);

многократное измерение размера детали ($n=2...30$) с использованием любого указанного выше метода измерения и запоминанием результатов;

измерение размеров партии (до 30) однотипных деталей с использованием любого указанного выше метода измерения и запоминанием результатов;

статистическую обработку серий измерений с запоминанием результатов;

разбраковку продукции по результатам как отдельных замеров, так и статистической обработки с индикацией на светодиодах состояний «Брак +», «Годен», «Брак —».

После выполнения серии измерений заданной длины в выходных параметрах прибора, начинающихся с номера 08, находятся полученные результаты: максимальное и минимальное значения в серии, среднее значение для серии, значение показателя формы, результаты последовательных измерений. Результаты могут быть считаны в пультовом режиме путем обращения к соответствующим выходным параметрам. Имеется также возможность последовательной индикации первых четырех указанных результатов в рабочем режиме.

Заключение

Объединение в базовом варианте достаточно универсальных аппаратной структуры и системной части программного обеспечения значительно упрощает создание новых типов программируемых приборов. При физической реализации разработанного прибора следует учитывать возможность распределения программного обеспечения между внутренним и внешним ПЗУ микросхем К586ВЕ1. Закрытый вариант исполнения УЦОО с размещением всего программного обеспечения во внутреннем ПЗУ, позволяющий сократить затраты на аппаратную часть, можно рекомендовать для приборов большой серийности. Открытый вариант исполнения УЦОО с размещением всего программного обеспечения во внешнем ПЗУ целесообразно использовать лишь в опытных образцах приборов. Для выпуска приборов малой серийности приемлем полуоткрытый вариант с размещением системного программного обеспечения во внешнем ПЗУ. Достоинство полуоткрытого варианта по сравнению с закрытым состоит в том, что переход к новому прибору можно осуществить простой сменой микросхем ПЗУ. В результате пользователь на одной аппаратной основе может иметь несколько различных программируемых приборов.

Статья поступила 11 марта 1985 г.

УДК 681.326

В. И. Воронин, К. В. Макаров, В. А. Старшова

КОНТРОЛЛЕР КЛАВИАТУРЫ НА БАЗЕ ОДНОКРИСТАЛЬНОЙ МИКРОЭВМ КМ1816ВЕ48

Контроллер предназначен для работы с клавиатурой и обеспечивает формирование кодов букв, знаков, цифр и команд в коде КОИ-7 или КОИ-8, а также вывод кодов данных на внешние устройства, имеющие интерфейс ИРПР СМ ЭВМ ОСТ 25778—77.

Питание контроллера клавиатуры (КК) осуществляется от источника постоянного тока напряжением +5 В ±5 %. Функциональная схема КК представлена на рис. 1 МикроЭВМ КМ1816ВЕ48 (DD1) содержит на кристалле 8-разрядный процессор, РПЗУ команд емкостью 1024 байт, ОЗУ емкостью 64 байт, 8-разрядный таймер-счетчик событий, интерфейс ввода-вывода. Программу РПЗУ можно изменить путем стирания содержимого ультрафиолетовым излучением с последующим электрическим программированием. Сопряжение микроЭВМ с матрицей клавиатуры, индикаторами режима работы клавиатуры, узлами звуковой сигнализации и интерфейса ИРПР осуществляет однокристальное программируемое устройство ввода-вывода параллельной информации (DD2).

В интегральной схеме DD3 запоминается адрес канала устройства сопряжения (DD2). Элементы DD5.1...DD5.5 микросхемы DD5 усиливают сигналы управления индикацией режима работы клавиатуры (НР — нижний регистр, ВР — верхний регистр, РУС — русские символы, ЛАТ — латинские символы и СУ — служебное управление). Устройство сопряжения с интерфейсом ИРПР (DD4, DD5.6, DD6 и VT2) обеспечивает согласование электрических параметров сигналов с каналом, формирует сигнал «Готовность источника» ($\overline{\text{ГИ}}$) и управление выводом сигналов «Строб» ($\overline{\text{СТР}}$) и «Контрольный разряд» (КР).

При подготовке контроллера к работе выполняются программирование ИС DD2, формирование исходных кодов режима работы и управления КК. Определение момента нажатия клавиши и защита от ложного срабатывания КК при дребезге контактов осуществляются программно. При формировании кода координаты нажатой клавиши проводится проверка на наличие/отсутствие одновременного нажатия нескольких клавиш.

Коды формируются методом многоэтапного преобразования, обеспечивающим выигрыш в емкости памяти в два раза по сравнению с ме-

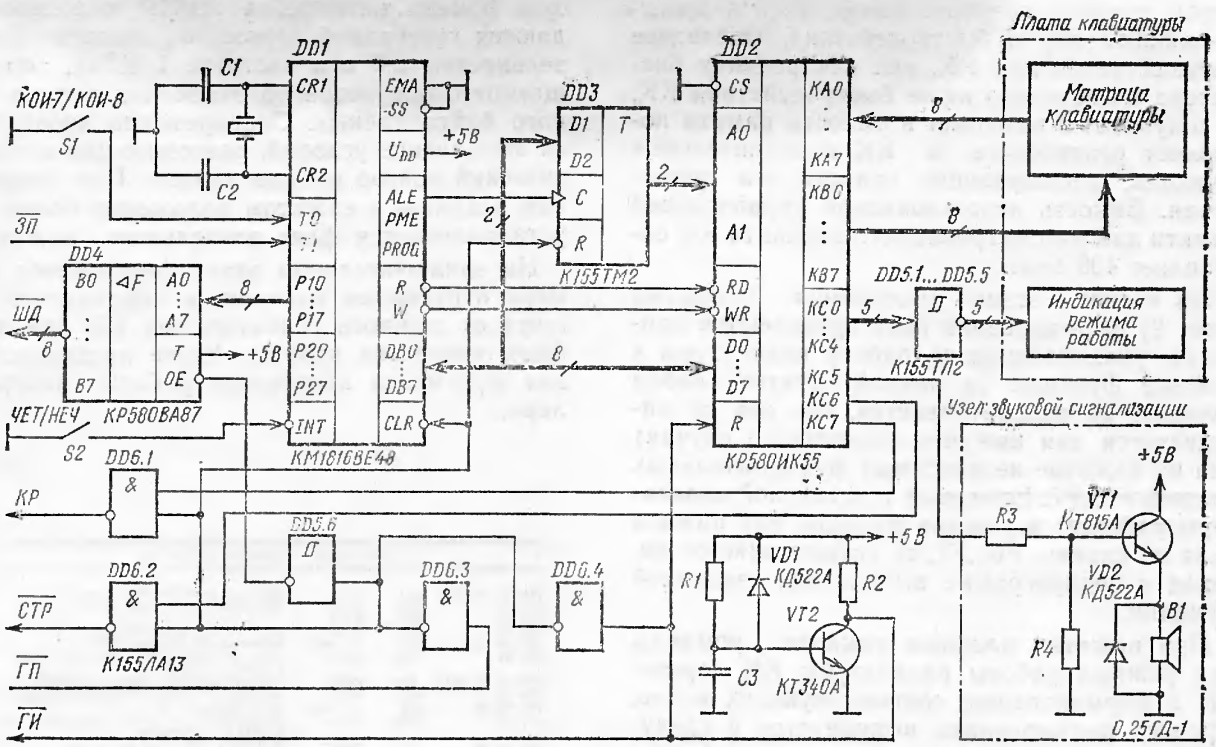
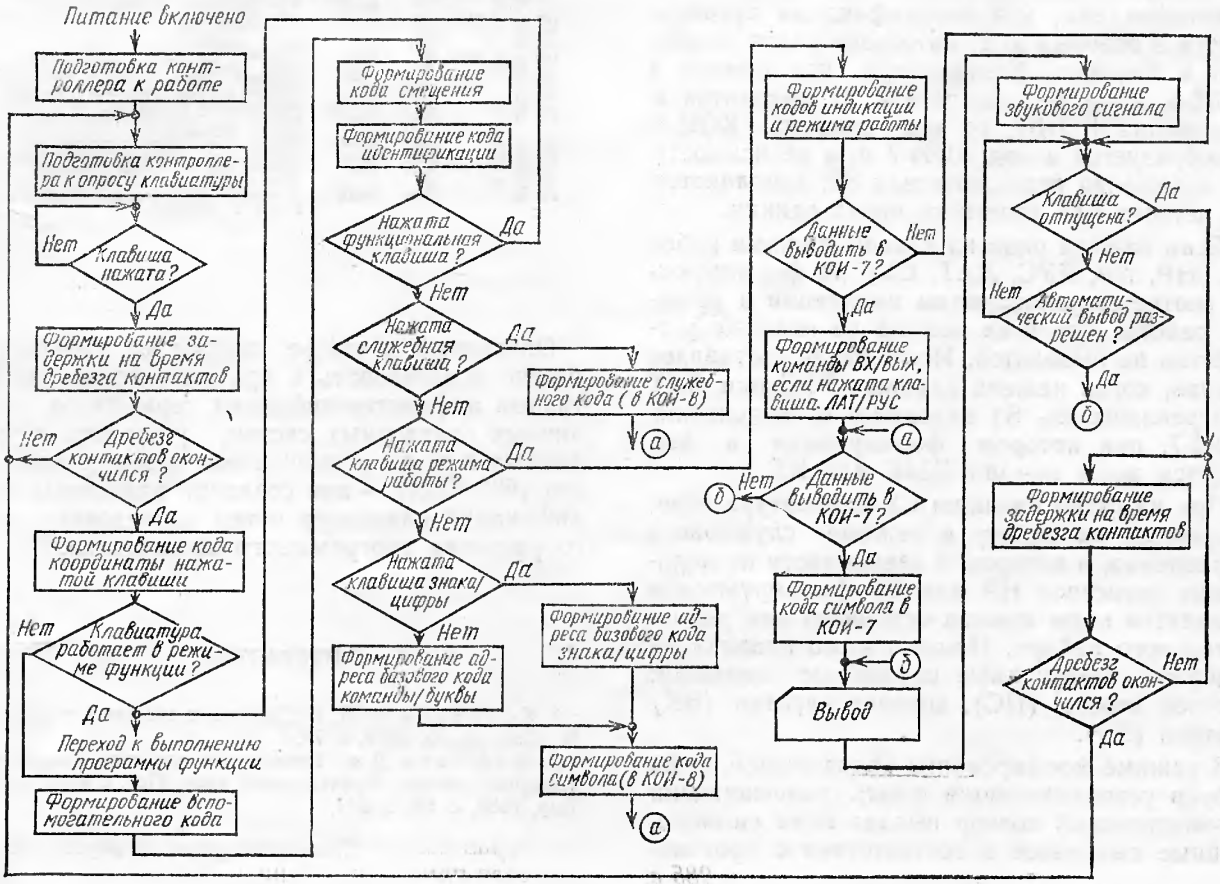


Рис. 1. Функциональная схема контроллера клавиатуры

Рис. 2. Алгоритм работы контроллера клавиатуры



тодом прямого преобразования, хотя и проигрывающим ему в быстродействии. Последнее несущественно для КК, так как реакция оператора значительно ниже быстродействия КК, а получаемый выигрыш в емкости памяти позволяет реализовать в КК дополнительные функции, расширяющие области его применения. Емкость использованной управляющей памяти для рассматриваемого варианта КК составляет 400 байт.

На первом этапе выполнения алгоритма (рис. 2) формирования кода проверяются признаки, указывающие на работу клавиатуры в режиме функции (в данной статье набор функций не рассматривается, так как он определяется для каждого конкретного случая) или на нажатие независимых функциональных клавиш F0...F7. Если поле телетайпной клавиатуры работает в режиме функции или нажата одна из клавиш F0...F7, то осуществляется переход к подпрограмме выполнения заданной функции.

При нажатии клавиши символа, команды или режима работы клавиатуры КК переходит к формированию соответствующих кодов. Процесс формирования выполняется в следующем порядке: код координаты нажатой клавиши преобразуется во вспомогательный код, по которому формируются коды смещения и идентификации; код идентификации преобразуется в базовый код; на основе кодов смещения и базового формируется код символ в КОИ-8. Если переключатель S1 находится в положении КОИ-7, то код символа КОИ-8 преобразуется в код КОИ-7 и, в зависимости от положения переключателя S2, дополняется до четного или нечетного числа единиц.

Если нажата одна из клавиш режима работы (НР, ВР, РУС, ЛАТ, СУ), то формируются соответствующие коды индикации и режима работы, код этих команд на внешнее устройство не выводится. Исключение составляет случай, когда нажата клавиша РУС или ЛАТ и переключатель S1 находится в положении КОИ-7, при котором формируются и выводятся коды команд ВЫХ или ВХ.

При нажатии клавиши СУ клавиатура переключается на работу в режиме служебного управления, в котором в зависимости от положения регистров НР или ВР формируются и выводятся коды команд основного или дополнительного набора. Помимо этого клавиатура содержит независимые служебные клавиши: перевод строки (ПС), возврат каретки (ВК) и забой (ЗБ).

В режиме формирования кодов знаков, цифр и букв устанавливается флаг, разрешающий автоматический повтор вывода кода символа. Данные выводятся в соответствии с протоколом обмена интерфейса ИРПР и сопровождаются генерацией звукового сигнала (длительностью 0,5 с и частотой 1 КГц), оповещающего оператора об окончании вывода одного байта данных. Одновременно проверяется выполнение условий, разрешающих автоматический повтор вывода данных. При удержании клавиши в нажатом положении более 1 с устанавливается флаг длительного нажатия.

На заключительном этапе фиксируется момент отпускания клавиши и реализуется защита от ложного срабатывания КК при дребезге контактов клавиш. Ниже представлены два фрагмента программы работы контроллера.

```
                ВЫВОД ДАННЫХ НА ВНЕШНЕЕ УСТРОЙСТВО

0EA 56 EA OUT:  JT1     OUT     ; ОЖИДАНИЕ НИЗКОГО УРОВНЯ НА ЛИНИИ ЗП
0EC FB          MOV     A,R3    ; ЗАГРУЗКА КОДА СИМВОЛА НА ШИНУ ДАННЫХ
0ED 39          OUTL    P1,A
0EE FE          MOV     A,R6    ; ЗАГРУЗКА В А КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
0EF 03 40       ADD     A, #40H ; ФОРМИРОВАНИЕ СИГНАЛА  СТР
0F1 91          MOVX    @R1,A   ; ЗАГРУЗКА КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
                                ; В ПОРТ С КР580ВК55
0F2 46 F2 OUT1: JNT1    OUT1    ; ОЖИДАНИЕ ВЫСОКОГО УРОВНЯ НА ЛИНИИ ЗП
0F4 91          MOVX    @R1,A   ; ЗАГРУЗКА В А КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
0F5 53 3F       ANL     A, #3FH ; СБРОС СИГНАЛОВ КР И СТР
0F7 91          MOVX    @R1,A   ; ВОЗВРАЩЕНИЕ КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
                                ; В ПОРТ С  КР580ВК55
0F8 24 00       JMP     SOUND   ; ПЕРЕХОД К ФОРМИРОВАНИЮ ЗВУКОВОГО СИГНАЛА

                ФОРМИРОВАНИЕ ЗВУКОВОГО СИГНАЛА

100 BD 04 SOUND:MOV     R5,#04H ; ЗАГРУЗКА В R4 И R5 КОДОВ, ОПРЕДЕЛЯЮЩИХ
102 BC FF   S1: MOV     R4,#0FFH; ПРОДОЛЖИТЕЛЬНОСТЬ ЗВУКОВОГО СИГНАЛА
104 23 FA FREQ: MOV     A, #0FAH; ЗАГРУЗКА КОДА ЧАСТОТЫ ЗВУКОВОГО СИГНАЛА
106 62          MOV     T,A     ; В РЕГИСТР ТАЙМЕРА
107 55          STRT    T       ; ЗАПУСК ТАЙМЕРА
108 16 0C WAIT: JTF     STOP    ; ПЕРЕХОД ПО ОКОНЧАНИИ ОТСЧЕТА ВРЕМЕНИ
10A 24 08       JMP     WAIT    ; ИНАЧЕ - ОЖИДАНИЕ
10C 65    STOP: STOP    TCNT    ; ОСТАНОВКА ТАЙМЕРА
10D 81          MOVX    @R1,A   ; ЗАГРУЗКА В А КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
10E D3 20       XRL     A, #20H ; ИНВЕРТИРОВАНИЕ СИГНАЛА ЗП
110 91          MOVX    @R1,A   ; ВОЗВРАТ КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ В
                                ; ПОРТ  С  КР580ВК55
111 EC 04       DJNZ    R4,FREQ ; ПЕРЕХОД ПО НЕНУЛЕВОМУ СОДЕРЖИМОМУ R4
113 ED 02       DJNZ    R5,S1   ; ПЕРЕХОД ПО НЕНУЛЕВОМУ СОДЕРЖИМОМУ R5
115 81          MOVX    @R1,A   ; ЗАГРУЗКА КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
116 53 DF       ANL     A, #0FH ; СБРОС СИГНАЛА  ЗП
118 91          MOVX    @R1,A   ; ВОЗВРАЩЕНИЕ КОДА УПРАВЛЕНИЯ ПЕРИФЕРИЕЙ
                                ; В ПОРТ  С  КР580ВК55
```

Описанный в статье контроллер клавиатур можно использовать и при разработке портативных алфавитно-цифровых терминалов, различных отладочных систем, устройств сбора данных и т. д., а имеющиеся резервы памяти его (600 байт) — при создании различных модификаций клавиатур путем последовательного развития программного обеспечения.

ЛИТЕРАТУРА

1. Соловейчик И. Е. Дисплеи в системах с ЭВМ.— М.: Сов. радио, 1979, с. 78.

2. Кофлрон Дж. Технические средства микропроцессорных систем. Практический курс: Пер. с англ.— М.: Мир, 1983, с. 99, с. 241.

Статья поступила 22 марта 1985 г.

УДК 681.322.042

Л. Н. Преснухин, Г. И. Фролов, Т. А. Куправа, В. С. Безобразов, В. А. Шахнов

УЧЕБНЫЙ КЛАСС НА ОСНОВЕ ДИАЛОГОВЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ

В средних школах г. Зеленограда созданы учебные вычислительные классы на основе диалоговых вычислительных комплексов типа «Электроника НЦ-80-20/1, 80-20/1М, 80-20/2М» (ДВК-1, ДВК-1М, ДВК-2М) и начата опытная эксплуатация. Выбор ДВК в качестве основного технического средства обусловлен следующими причинами:

в настоящее время налажен массовый выпуск этого класса ЭВМ;

ДВК программно совместимы с самыми массовыми в стране мини- и микроЭВМ серии «Электроника», что особенно важно в профориентации учащихся;

ДВК имеет стандартную клавиатуру и специальный дисплейный монитор, обеспечивающий приемлемое для школьного обучения качество изображения;

ДВК обеспечивает диалоговый режим общения с обучаемым, что способствует быстрому и качественному закреплению приобретенных знаний.

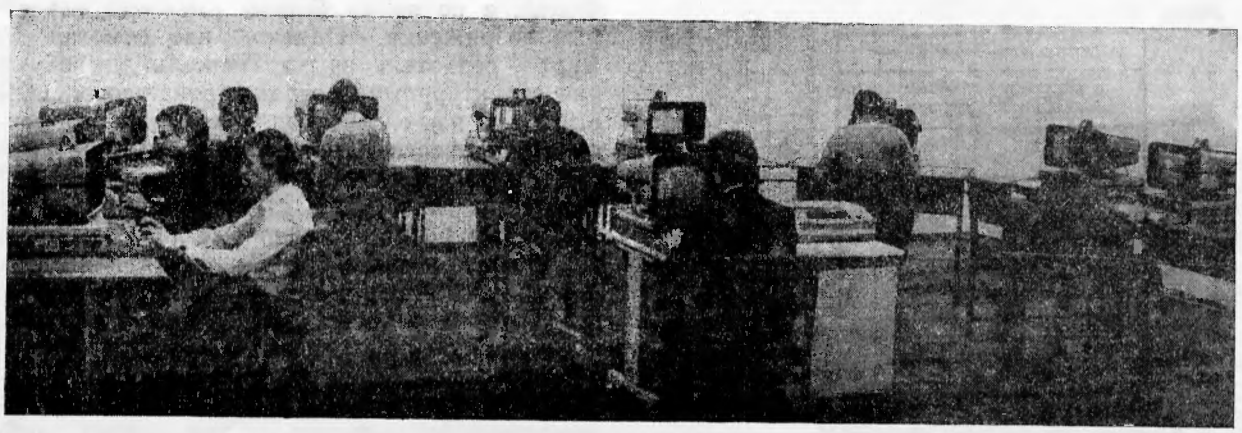
В состав ДВК-1 (ДВК-1М) входят микроЭВМ «Электроника МС 1201.01», алфавитно-цифровой дисплей и клавиатура. В состав ДВК-2М, кроме того, входят накопитель на гибких магнитных дисках и алфавитно-цифровое устройство.

МикроЭВМ построена на базе СБИС однокристального 16-разрядного микропроцессора К1801ВМ1 и выполнена на одной плате, включающей в себя: ОЗУ (56К байт); контактное устройство для БИС ПЗУ (8К байт); параллельный байтовый интерфейс; последовательный интерфейс; интерфейс НГМД.

Для аудиторной организации обучения разработана простая локальная сеть, в которой ДВК-1 связаны линиями последовательной передачи информации с одной главной ЭВМ ДВК-2М (рис. 1). По общей идеологии сети все функции управления и обмена программами сосредоточены на главной ЭВМ. Связь с ДВК-1 устанавливается преподавателем, находящимся за главной ЭВМ, только на момент обмена информацией. Он может вмешаться в работу учащегося, загрузить или выгрузить файл с диска своей машины в микроЭВМ учащегося, просмотреть, скорректировать, выполнить программу в микроЭВМ учащегося с пульта главной ЭВМ и передать ему управление. Имеется возможность передачи из микроЭВМ учителя в ДВК-1 для исполнения программ, написанных на языке ассемблер, например, интерпретатор языка Бейсик, компилятора языка Квейсик.

Организация подобного класса дает возможность оперативной связи учитель — ученик, исключает несанкционированный доступ учащихся к общей файловой системе главной ЭВМ, позволяет уменьшать затраты на одного учащегося, так как в классе можно иметь один НГМД.

Аппаратную поддержку связи главной ЭВМ с ДВК-1 обеспечивают две платы контроллера телеграфного канала. Каждая из них обслуживает связь по шести последовательным линиям. В ДВК-1 для связи с главной ЭВМ используется имеющийся последовательный интерфейс, который переключается в «линию» с помощью пульта преподавате-



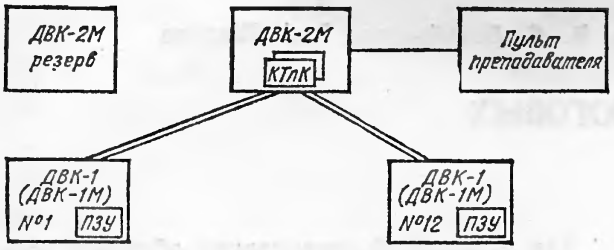


Рис. 1. Схема учебного класса

ля для дистанционной ручной установки связи. Соединение ДВК-1 с главной ЭВМ осуществляется кабелем из четырех витых пар и узлов оптоэлектронной развязки (оптронных приемников К293ЛП1А и оптронных передатчиков АОТ110А) по линиям канала типа «20 мА токовая петля». Отличительная особенность аппаратной части сети — доступная стоимость, простота монтажа и эксплуатации.

В системное программное обеспечение главной ЭВМ входят программы приема-передачи файлов, организации диалога с ДВК-1 управления файлами на НГМД в операционной системе ДВК.

Системное программное обеспечение ДВК-1 реализовано «в кремнии» в виде БИС ПЗУ К1801РЕ1-058. Применение в учебной микроЭВМ «кремниевых» программных средств позволяет получить надежность, характерную для аппаратных схем, обеспечить простоту обслуживания и неизменность интерфейса между пользователем и микроЭВМ, в то же время сохранить гибкость, присущую программным средствам. Указанная микросхема ПЗУ емкостью 8К байт содержит интерпретатор диалогового языка программирования и программные модули для поддержания обмена информацией с главной ЭВМ по последовательному каналу. Микросхема ПЗУ устанавливается в имеющееся на плате микроЭВМ контактное устройство и может быть заменена в случае разработки новой микросхемы. Система резидентна в ПЗУ и функционирует в нем, не требуя перезаписи в оперативную память и используя в ней лишь 72 ячейки для системных целей. Пользователю предоставляется около 48 К байт оперативной памяти для программ и данных (рис. 2).

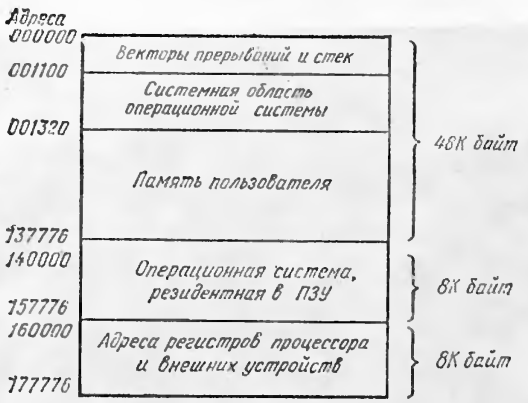


Рис. 2. Адресное пространство микроЭВМ

При реализации программных средств и выборе языка авторы руководствовались тем, что следует различать начальное обучение программированию и изучение программирования как некоторой систематической дисциплины, [1, 2]. К языкам структурного программирования относятся Алгол, Паскаль, Ада, Си и др. Многие концепции этих языков отражают последние достижения в методологии программирования и предназначены больше для создания крупных программных систем. Но начинать с них изучение программирования крайне затруднительно.

Для понимания такого языка, как, например, Паскаль, требуется знание основных семантических понятий программирования. Кроме того, для поддержания транслятора языка структурного программирования необходимы внешние запоминающие устройства типа НГМД и т. п.

Для начального обучения программированию наиболее приспособлены диалоговые языки программирования типа Бейсик, Фокал. Особого внимания заслуживает графический язык Лого, ориентированный на процедуры школьного обучения. В отечественной практике используется учебная система «Школьница» с языками «Рапира» и «Робик» [4].

Авторами статьи был выбран язык Фокал, являющийся стандартным международным языком серии мини- и микроЭВМ. Отсутствие в ДВК-1 НГМД для хранения системного программного обеспечения и ограниченность в объеме ПЗУ (8К байт) для размещения системы оказали решающее влияние на выбор именно компактного для реализации языка Фокал. Выбор менее распространенного языка Фокал, а не языка Бейсик вызывает некоторые возражения. Однако, как отметил Н. Вирт — создатель языка Паскаль для обучения структурному программированию: «...если выбирать язык обучения только по его распространенности и доступности, мы обрекаем себя на застой, поскольку язык, которому мы больше всего обучаем, становится из-за этого еще более распространенным» [3]. Язык Фокал близок к языку Бейсик, легок для самообучения, позволяет писать компактные программы, дает простой доступ к необходимым вычислительным ресурсам, обеспечивает простой переход к «производственным», распространенным на практике языкам, и в составе минимальных технических средств решает проб-

лему обучения основам программирования [5, 6].

Интерпретатор языка Фокал реализует расширенную версию, совместимую со стандартной версией языка Фокал, включает полный набор необходимых средств для изучения основ программирования, составления эффективных обучающих, игровых программ, построения учебных систем управления экспериментом. Максимальное удовлетворение требованиям учебной среды потребовало добавления в интерпретатор следующих новых средств: оператора экранного редактирования программных строк; выдачи полных текстовых диагностических сообщений об ошибках; наиболее часто встречаемых в практике математических функций; операторов выдачи справочной информации и запуска встроенной тестовой задачи; средств управления прохождением программы, средств повышения надежности и живучести интерпретатора.

Интерпретатор включает в себя также пакет из десяти графических функций: управление графическим экраном, графическим курсором, режимом вывода; построением окружностей, прямоугольников, векторов; выводом точек в абсолютных и относительных координатах. Эти функции работают при наличии специально разработанного для ДВК-1М контроллера графического вывода информации. Контроллер графики собран на полуплате микроЭВМ «Электроника 60», устанавливается в ее системную магистраль и обеспечивает формирование поля из 286×400 черно-белых точек на весь экран (16К байт). Он содержит три регистра, программно-доступных по записи и чтению на магистрали: регистр управления РУ для задания режимов работы контроллера; регистр адреса РА, в который записывается адрес обрабатываемого элемента изображения (байта) в ОЗУ; регистр данных РД для записи и считывания байта графической информации из ОЗУ контроллера графики по адресу, указанному в РА. Важность наличия графических средств в учебных микроЭВМ не вызывает сомнений.

Описанный учебный класс используется для обучения основам программирования и различным предметам с помощью обучающих программ. Преподавание информатики и основ программирования в школах ведется учителями, прошедшими 50-часовую подготовку при МИЭТ. Обучающие программы на языке Фокал с использованием графических средств по различным предметам школьной программы (физика, математика, русский и английский языки, история и т. п.) разрабатываются специалистами кафедры «Вычислительная техника» МИЭТ в сотрудничестве с учителями школ. Опыт эксплуатации подтвердил эффективность применения учебных классов на основе ДВК.

ЛИТЕРАТУРА

1. Дал О., Дейкстра Э., Хоар А. Структурное программирование: Пер. с англ.— М.: Мир, 1975.—247 с.

2. Хьюз Дж., Мичтом Дж. Структурный подход к программированию: Пер. с англ.— М.: Мир, 1980.— 278 с.

3. Йенсен К., Вирт Н. Паскаль. Руководство для пользователя и описание языка: Пер. с англ.— М.: Финансы и статистика, 1982.—151 с.

4. Звенигородский Г. А., Глаголева Н. Г., Земцов П. А., Налимов Е. В., Цикоза В. А. Программная система «Школьница» и ее реализация на персональных ЭВМ.— Микропроцессорные средства и системы, 1984, № 1, с. 50—55.

5. Фролов Г. И., Горовой В. Р., Куправа Т. А. Интерпретатор языка высокого уровня для индивидуальной микроЭВМ.— В кн.: Алгоритмы и программы системного математического обеспечения ЕС ЭВМ и кросс-средств. Сб. научных трудов. М.: МИЭТ, 1982, с. 57—61.

6. Фролов Г. И., Куправа Т. А., Чанров Ю. Г., Капустина Г. Н. Программирование на языке Фокал. Учебное пособие / Под ред. Л. Н. Преснухина.— М.: МИЭТ, 1984.—95 с.

Статья поступила 29 марта 1985 г.

ИНФОРМАЦИЯ

В 1986 г. Международный центр научной и технической информации (МЦНТИ) издаст комплект документации по языку программирования Бейсик/F в серии «Методические материалы и документация по пакетам прикладных программ».

Издание будет содержать руководство для пользователей, описания языка Бейсик/F, интерпретатора и компилятора, средств настройки на аппаратные особенности микроЭВМ.

Ориентировочная цена 2 р. 40 к.

Заказы на комплект документации принимаются в СССР магазином № 93 «Книга-почтой» Москниги по адресу: 117168, Москва, В-168, ул. Кржижановского, д. 14, корп. 1.

БИНТИ № 1 (2194)

Фирма «Интел корпорейшн» (США) разработала ЗУ на цилиндрических магнитных доменах, конструктивно выполняемое в виде съемной кассеты емкостью 1 М байт.

Новое ЗУ может использоваться в различном регистрирующем оборудовании для сбора и накопления данных, которые затем могут быть обработаны, например, в учрежденческой ЭВМ. ЗУ превосходит накопители на свободно плавающих магнитных дисках по быстродействию, надежности работы и прочности конструкции, а полупроводниковым ЗУ оно уступает по быстродействию. Новое ЗУ рассчитано на практическое применение в рабочем диапазоне температур от 0 до 65 °С и имеет стоимость 605 долл. при объеме партии 100 шт.

№ 29458, 26 октября 1984 г., с. 30

УДК 681.326+681.325.5

П. А. Семенов

МИКРОКОНТРОЛЛЕРЫ НА БАЗЕ БИС КР580 ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60» «ЭЛЕКТРОНИКА НЦ-80-01Д»

Построенные на базе микропроцессорного комплекта БИС серии КР580 контроллеры системного терминала, графического дисплея и графопостроителя для микроЭВМ «Электроника 60» и «Электроника НЦ-80-01Д» значительно расширяют функциональные возможности микроЭВМ, повышают производительность системы, упрощают системное программное обеспечение (ПО). После простейшего перепрограммирования эти контроллеры можно адаптировать под любое периферийное устройство, не изменяя системное ПО. Это позволяет:

повысить комплексную производительность системы благодаря распараллеливанию задач ввода-вывода и обработки данных;

разработать «периферийно-независимые» программы и операционные системы, используя стандартный программный интерфейс ввода-вывода;

локализовать интерфейсные сервисные функции (например, «человек — машина») на уровне самого интерфейса и, значительно сократив объем системного программного обеспечения, повысить вычислительные возможности системы.

Контроллер системной консоли

Контроллер системной консоли (КСК) предназначен для двунаправленной связи микроЭВМ с любым дисплеем и функционирует в рамках операционной системы (ОС) RT11.

В число функций КСК входят:

управление четырьмя группами по 8 непрограммируемых функциональных клавиш с основными командами монитора ОС RT11, отладчиков и интерпретаторов, разбитых на четыре группы в зависимости от языков программирования (Паскаль, Фортран, Бейсик и макроассемблер);

программирование и управление двумя группами по 8 программируемых функциональных ключей с вызовами, ответами и командами пользователя, необходимыми для адаптации к конкретным задачам;

управление стеком команд;

распознавание и перекодировка командных последовательностей экранного редактора К52 (KED), а также редактора командной строки SL ОС RT11;

распознавание и обработка командных последовательностей терминала, работающих в «локальном» режиме (при просмотре содержимого экрана и т. п.);

управление графикой на уровне макрокоманд графического монитора (для графических терминалов).

Основа структурной схемы КСК (рис. 1) — это микропроцессор КР580ИК80А [1]. Он связан через системный контроллер КР580ВК28 и магистраль данных D^s с программируемым параллельным интерфейсом (ППИ), таймером (Т) и блоком приоритетных прерываний (БПП).

КСК обменивается информацией с каналом «Электроника 60» через дуплексную пару регистров RG_1—RG_2 и приемопередатчики F_1 интерфейсной платы И5 параллельного ввода-вывода. Управление обменом со стороны МП осуществляется в режиме прерываний для RG_1 и в программном режиме для RG_2 с маскированием соответствующего входа запроса в БПП.

Чтобы исключить ложные ситуации при обработке прерываний, в БПП введено специальное прерывание регенерации статистических запросов REFR (период ~30 мс), конъюнктирующее все «активные» на данный момент запросы с импульсным сигналом $\overline{REFR}$.

В зависимости от типа интерфейса КСК связывается с внешним терминалом либо через последовательный канал RS-232C с помощью универсального синхронно-асинхронного приемопередатчика (УСАПП)

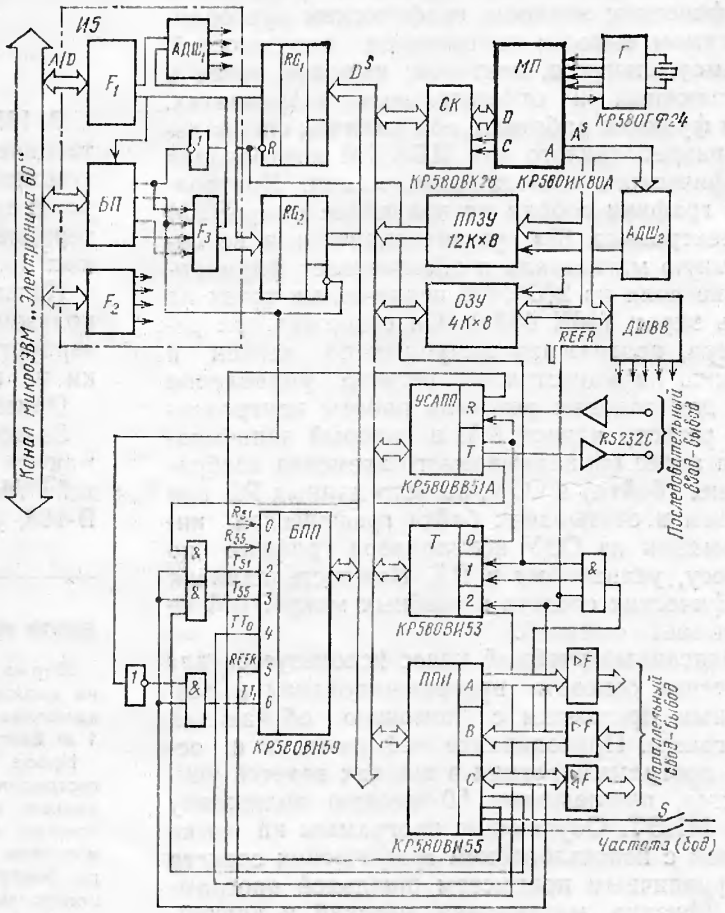


Рис. 1. Микроконтроллер системной консоли:

F_1, F_2, F_3 — формирователи; БП — блок прерываний; $АДШ_1$, $АДШ_2$ — адресные дешифраторы; МП — микропроцессор; Г — генератор; СК — системный контроллер; ППЗУ — программируемое постоянное запоминающее устройство; ОЗУ — оперативное запоминающее устройство; ДШВВ — дешифратор ввода-вывода; БПП — блок приоритетных прерываний; УСАПП — универсальный синхронно-асинхронный приемопередатчик; Т — таймер; ППИ — программируемый параллельный интерфейс

КР580ВВ51А, либо через параллельный канал с помощью ППИ типа КР580ИК55, работающего в режиме 1.

Скорость обмена по каналу RS-232C программируется таймером T0 в зависимости от состояния переключателя S. Для согласования скорости передачи данных в микроЭВМ с быстродействием системного драйвера TT: ОС RT11 используется таймер T1. Интерфейсы переключаются программно.

Программное обеспечение КСК — это фоново-оперативная операционная система реального времени. Она располагается в ППЗУ (12 Кбайт).

Фоновая задача (рис. 2) осуществляет ввод (и обработку) данных из канала микроЭВМ в контроллер через регистр RG_2 (сигнал готовности RG_2 в БПП замаскирован и опрашивается программно), а также вывод данных в кольцевой буфер TRB для последующей передачи в терминал. Фоновая задача распознает управляющие последовательности, соответствующие входу (SET EDIT) в экранный редактор K52 ОС RT11 и выходу (EXIT EDIT) из него, а также команды разрешения (SL ON) и запрета (SL OFF) редактирования командной строки монитора (драйвер SL ОС RT11 v05). В первом случае происходит перезагрузка функциональных клавиш, во втором — устанавливается бит локального режима управляющих функций. Это запрещает передачу их кодов в микроЭВМ монитором приемника терминала и разрешает пересылку в фоновую задачу. Распознавание и перекодировка команд управления курсором позволяют адаптировать экранные редакторы K52 (KED) и SL ОС RT11 под любой терминал. Коды распознаются последовательно, начиная с управляющего символа ($<ESCAPE>=033_8$).

В *оперативном режиме* могут находиться до четырех задач одновременно:

монитор приемника терминала RT (наивысший приоритет 0 или 1);

передатчик терминала TRT (приоритет 2 или 3);

передача в канал CTT (приоритет 4);

регенерация статических запросов прерываний REFR (приоритет 5). Это позволяет совместно с решением фоновой задачи обеспечить двунаправленный обмен и преобразование информации в режиме реального времени.

Фоновая задача связана с оперативной задачей TRT (рис. 3) через кольцевой буфер данных TRB (256 байт), флаг состояния TRBE (TRB пуст) и секцию инициализации, снимающей маску запроса передатчиков УСАПП или ППИ (T_{51} или T_{55}) в БПП и сбрасывающей бит TRBE. Данные выводятся в передатчик по прерываниям до совпадения указателя считывания кольцевого буфера TRBR с указателем записи TRBW с последующей установкой маски в БПП и бита TRBE. Использование кольцевого буфера TRB позволило совместить процессы записи и считывания для него, что эквивалентно применению двух дублирующих буферов.

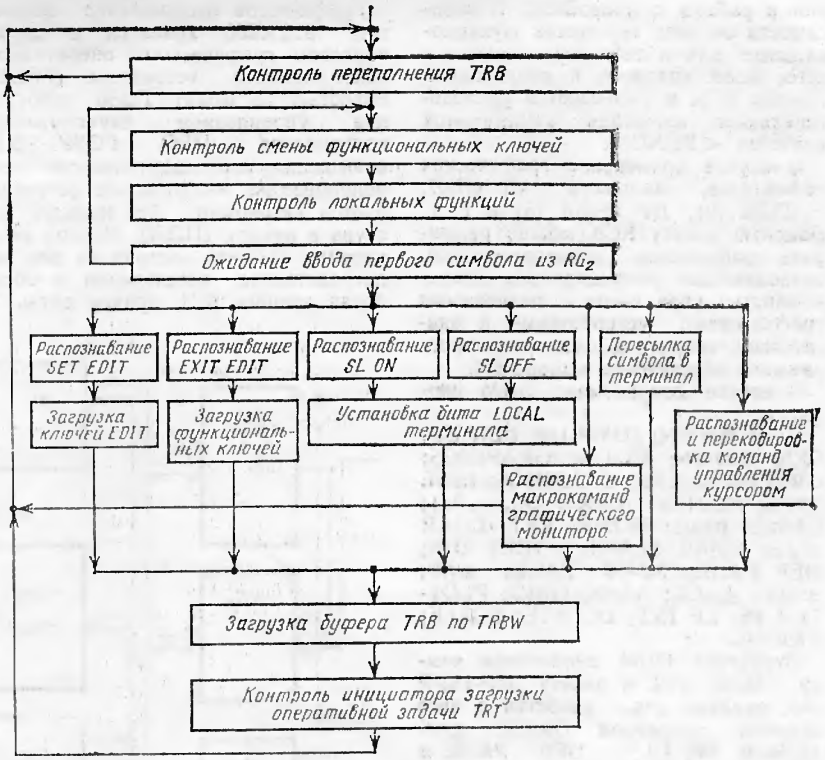


Рис. 2. Структура фоновой задачи контроллера системной консоли

Монитор RT всегда активен в системе и имеет наивысший приоритет среди всех оперативных задач. В число его функций входят:

распознавание и перекодировка командных управляющих последовательностей в режимах EDIT (K52) и SL;

распознавание и передача фоновой задаче кодов локальных (выполняемых в пределах интерфейса и терминала) управляющих функций;

распознавание кодов функциональных клавиш и инициализация соответствующего блока фоновой задачи;

управление программированием функциональных клавиш пользователя;

инициализация задачи CTT.

Командные и локальные управляющие последовательности распознаются монитором так же, как фоновой задачей, с той разницей, что в данном случае используется символьный стек для анализа предыдущих кодов в многосимвольной последовательности. Перекодированные данные поступают либо в кольцевой буфер приемника RB (аналогичен TRB) с последующей инициализацией задачи CTT (аналогична TRT, рис. 3), либо в обменник фоновой задачи, состоящий из буфера данных и битов инициализации/завершения соответствующих функций (см. рис. 2).

В число локальных функций входят команды HOME, стирания экрана и просмотра памяти.

Частный случай локальных функций — функциональные клавиши. Они позволяют значительно упростить процессы программирова-

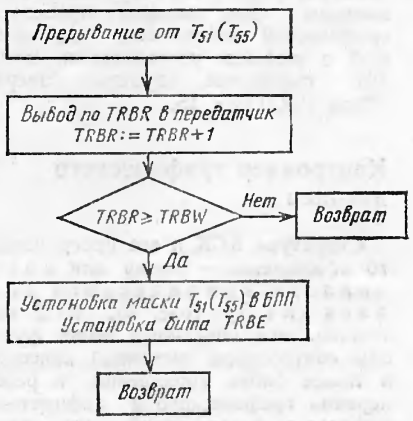


Рис. 3. Оперативная задача TRT

ния и работы с микроЭВМ. В зависимости от типа терминала функциональные ключи либо пересылаются в него, либо хранятся в оперативной памяти КСК и вызываются последовательным нажатием эмулируемой клавиши <FUNC>.

В случае применения графических терминалов, например VT 47607, ЭПГСМ [2], HP 2623A [3] и т. д., фоновую задачу КСК можно расширить графическим монитором GDM, позволяющим унифицировать макрокоманды управления различными графическими устройствами и значительно сократить системное программное обеспечение микроЭВМ.

В состав макрокоманд GDM входят:

— JUMP ON; JUMP UP; PEN UP; PEN DOWN; CLEAR GRAPHICS; GRAPHICS; EXIT GRAPHICS; LINE TYPE; PLOT XY; DRAW {A_1, ..., A_n}; LABEL {text}; HOME; RET; CHAR SIZE; CHAR SLANT; TEXT DIR; DEF PAGE; JUMP PAGE; ±nB; ±nLF; ±nUL; AXES; GRID; PLOTTER IS; LF INT; LOCATE; SCALE; FRAME.

Структура GDM аналогична языку Basic AGL и пакету HP-ISPP [4], однако для удобства в него введены операторы DRAW, ±nK (K = B, LF, UL)), DEF PAGE и JUMP PAGE.

Так, DRAW {A_1, ..., A_n} с помощью последовательности узловых точек {A_i} позволяет описать любую детерминированную графическую структуру и заменяет соответствующее число операторов MOVE. Длина последовательности A_i не ограничена. Оператор ±nK позволяет сформировать n пробелов (B), переводов строки (LF) или подчеркиваний (UL) в любом относительно текущего положения курсора направлении. Операторы DEF PAGE и JUMP PAGE пространственно разъединяют графическую и текстовую зоны, что удобно для создания текстовых комментариев и т. п.

Таким образом, составление программы, манипулирующей с графикой, сводится лишь к стандартным макровызовам (или вызовам процедур) графической библиотеки, работающей с любыми устройствами, которые сменяются простым оператором PLOTTER IS.

Контроллер графического дисплея

Структура КСК и его программного обеспечения — основа для контроллера графического дисплея (КГД) (рис. 4). КГД выполняет все описанные выше функции контроллера системной консоли, а также блока управления и регенерации графического и алфавитно-цифрового изображений для стандартного монитора.

Графическое изображение форматом 512×256 хранится в двухпортовом графическом оперативном запоминающем устройстве (ГОЗУ) емкостью 16 Кбайт. ГОЗУ работает под управлением двухпортового контроллера (ДПК) ГОЗУ. Для повышения производительности микропроцессора изображение регенерируется автономно без прямого доступа в память (ПДП). Период регенерации (1 мкс) состоит из фаз непосредственно регенерации и обработки запроса МП, причем связь с МП синхронизируется сигналом ожидания $\overline{NRDY}$.

Алфавитно-цифровая информация из ОЗУ контроллера регенерируется через канал ПДП под управлением контроллера алфавитно-цифрового дисплея КАЦД КР580ВГ75 и контроллера прямого доступа в память КПДП КР580ВТ57. Работа КАЦД и ДПК ГОЗУ синхронизируется блоком синхронизации БС.

Особенность *программного обеспечения КГД* — отсутствие процесса передачи информации из него во

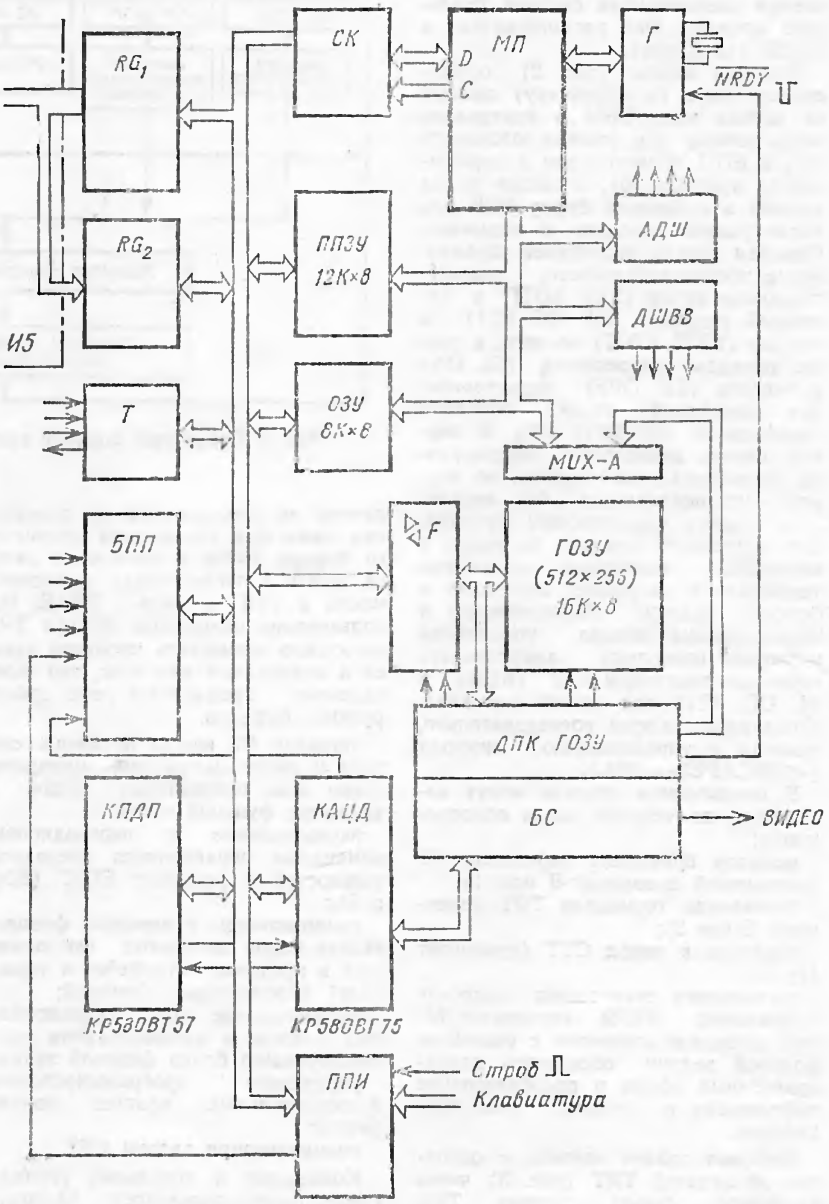


Рис. 4. Контроллер графического дисплея:

MUX—A — мультиплексор адреса; ◁ F ▷ — приемопередатчик; ГОЗУ — графическое ОЗУ; ДПК ГОЗУ — двухпортовый контроллер ГОЗУ; БС — блок синхронизации; КПДП — контроллер прямого доступа в память; КАЦД — контроллер алфавитно-цифрового дисплея. Остальные обозначения те же, что и на рис. 1

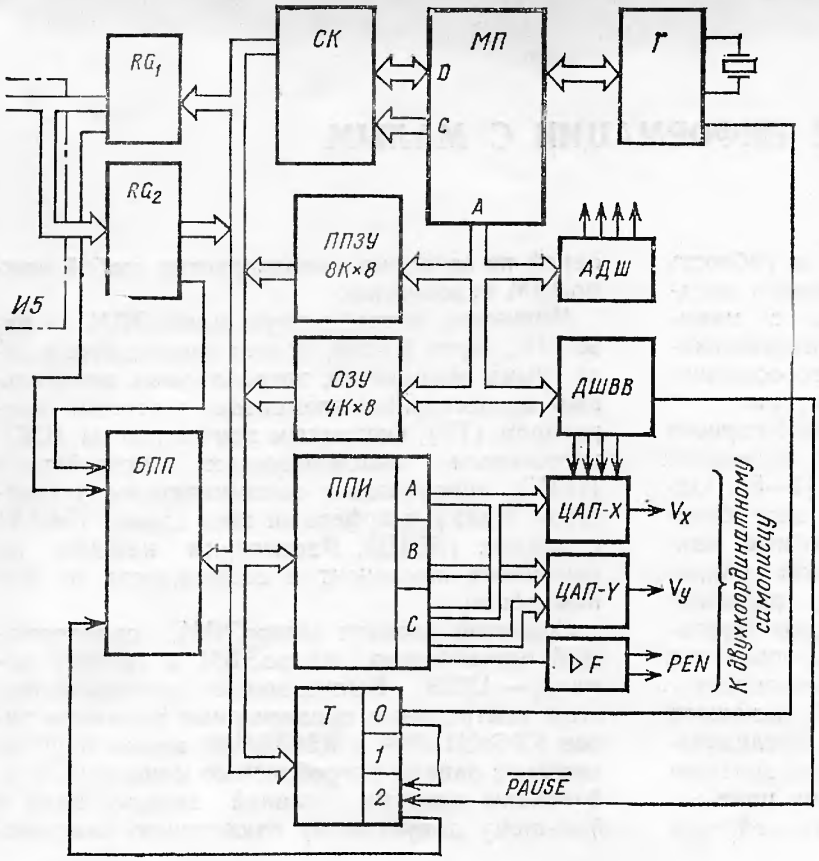


Рис. 5. Контроллер графопостроителя.
Обозначения те же, что и на рис. 1

внешнее устройство. Кроме удаления из системы оперативной задачи TRT, возможно перераспределение функций приема, обработки и выполнения инструкций интерфейса И5 между оперативными и фоновой задачами. В связи с большим временем выполнения графических макрокоманд GDM наиболее оптимален (с точки зрения комплексного быстродействия микроЭВМ) прием данных в кольцевой буфер (2 Кбайт) из интерфейса в оперативном режиме с последующей инициализацией фоновой задачи их обработки и выполнения.

Контроллер графопостроителя

Другой вариант модификации КСК для графических устройств — контроллер графопостроителя (КГ) (рис. 5), реализованного на основе двухкоординатного самописца. Координатные напряжения V_x и V_y формируются цифроаналоговыми преобразователями ЦАП—X и ЦАП—Y, а сигнал управления пером-формирователем F с выходов параллельного интерфейса КР580ИК55. Разрядность ЦАП выбирается исходя из максимально возможной точности двухкоординатных самописцев и составляет 10 бит для лучших образцов.

Программное обеспечение КГ аналогично ПО КГД и включает в себя фоновый графический монитор GDM и две оперативные задачи ввода-вывода данных из интерфейса И5 через дуплексную пару RG_1—RG_2.

Наличие инерционности графопостроителя при переводе каретки и управлении пером привело к необходимости оптимизации времени выполнения T_B оператора JUMP, (основа многих макрокоманд графического монитора).

Временные задержки выполнения команд JUMP (5 ... 150 мс), PEN UP/DOWN (100 мс) и пошагового выполнения PLOT (2 ... 7 мс) формируются программируемым таймером T.

Емкость кольцевого буфера фоновой задачи КГ (3,5 Кбайт) практически полностью исключает задержки в работе микроЭВМ, вызванные передачей данных в контроллер при синтезе сложных графических структур.

К достоинствам описанных микроконтроллеров следует отнести и возможность адаптации под любую микроЭВМ без изменения внутренней структуры благодаря стандартному параллельному асинхронному протоколу обмена с интерфейсом канала.

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.—272 с.

2. Система малых электронных вычислительных машин СМ ЭВМ. Каталог 1.— Союзаграннрибор, 1982.

3. Electronic instruments and systems. Hewlett-Packard, 1983.

4. Graphics ROM for HP9845. Hewlett-Packard Desktop Computer Division, 1981.

Статья поступила 5 февраля 1985 г.

БИНТИ № 3 (2196)

В последнее время наблюдается расширение комплектов команд в коммерческих ЭВМ, что, с одной стороны, обусловлено стремлением программистов расширить диапазон выполняемых ЭВМ функций, а с другой стороны, более высоким быстродействием логических схем по сравнению с быстродействием ЗУ.

Однако, с созданием микроузлов с высоким и очень высоким уровнем интеграции различия в быстродействии логических схем и ЗУ исчезают, и специалисты Калифорнийского технологического института ввели специальные команды повтора, которые позволяют использовать несколько команд для одной и той же группы данных и сократить общее число исполняемых ЭВМ команд.

Так, фирма «Интернэшнл бизнес машинз корпорейшн» разработала ЭВМ «801» типа «RISC», которая при сокращенном комплекте команд сравнима по характеристикам с ЭВМ «ИБМ-370», а фирма «Белл лабораториз» (отделение фирмы «Американ телефон энд телеграф») создали ЭВМ, в которой число команд сокращено до 40, тогда как в ЭВМ фирмы «Интернэшнл бизнес мэшинз корпорейшн» с аналогичными характеристиками используются 256 команд. Анализ показал, что при исполнении программ постоянно используется, как правило, лишь 30—40 команд.

УДК 681.326

Н. А. Тарасов

УСТРОЙСТВО ОБРАБОТКИ ИНФОРМАЦИИ С МАЛЫМ ЭНЕРГОПОТРЕБЛЕНИЕМ

Функциональные возможности и гибкость аппаратуры можно расширить, широко внедряя микропроцессорные устройства с минимальными весо-габаритными характеристиками и потребляемой мощностью. Это особенно важно для малогабаритной аппаратуры.

Минимальный набор микропроцессорного комплекта БИС серии КР580 [1, 2] позволяет реализовать функции микроЭВМ [1—5]. Однако из-за сравнительно высокой потребляемой мощности использовать подобные микроЭВМ в переносной малогабаритной аппаратуре с аккумуляторным питанием, работающей в полевых условиях, очень трудно. Поэтому проанализируем возможность создания устройства на основе центрального процессорного элемента (ЦПЭ) серии КР580, имеющего минимальное энергопотребление и предназначенного для встраивания в малогабаритную аппаратуру и для работы в жестких полевых условиях. При этом снижение быстродействия одной из основных характеристик любой микроЭВМ недопустимо.

Мощность, потребляемую микроЭВМ на базе БИС серии КР580, можно представить в виде суммы мощностей, потребляемых центральным процессорным элементом, тактовым генератором (ТГ), системным контроллером (СК), постоянным запоминающим устройством (ПЗУ), оперативным запоминающим устройством (ОЗУ) и буферами шин адреса (БША) и данных (БШД). Рассмотрим каждую из указанных компонент и возможности их минимизации.

Основной элемент микроЭВМ, определяющий организацию микроЭВМ и систему команд, — ЦПЭ. Выпускаемые промышленностью центральные процессорные элементы типов КР580ИК80А и К580ВМ80, имеют приблизительно равные потребляемые мощности. Требованиям жестких условий эксплуатации и большему допустимому отклонению напряже-

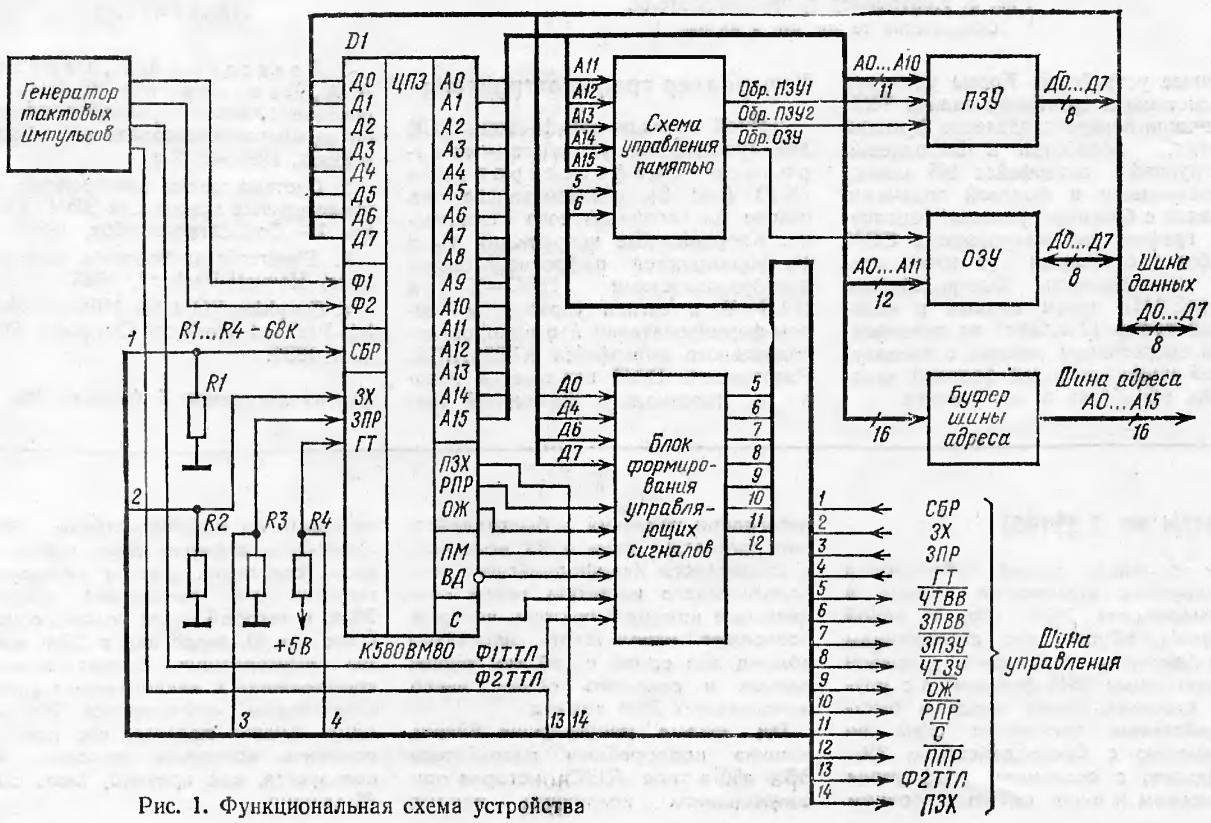


Рис. 1. Функциональная схема устройства

ний источников питания от номинального значения (±10 %) наиболее полно удовлетворяет ЦПЭ типа К580ВМ80. Итак, мощность, потребляемая ЦПЭ, строго определенна и не может быть уменьшена. Она зависит от мощности, потребляемой К580ВМ80, и равна 0,75 Вт.

Все остальные элементы устройства не определяют его организации и системы команд, поэтому их оптимизация — главный резерв снижения потребляемой мощности.

Интегральные микросхемы КМДП-структуры [6] и маломощные ТТЛШ, например серии К533, в значительной степени решают поставленную задачу.

Функциональная схема устройства с малой потребляемой мощностью (рис. 1) включает в себя генератор тактовых импульсов, ЦПЭ, схему управления памятью, ПЗУ, ОЗУ, блок формирования управляющих сигналов и буфер шины адреса. Здесь A1...A15 — выходы шины адресов, а D0...D7 — входы-выходы шины данных ЦПЭ, Ф1 и Ф2 — входы тактовых сигналов. ЦПЭ имеет также входные и выходные сигналы управления. Входные сигналы управления (СБР — сброс, ЗХ — захват шин, ЗПР — запрос прерывания и ГТ — готовность данных к передаче) поступают с разъема устройства. Выходные сигналы управления ($\overline{РПР}$ — разрешение прерывания, $\overline{ОЖ}$ — ожидание, $\overline{ПМ}$ — разрешение приема информации через шину данных в ЦПЭ, $\overline{ВД}$ — признак выдачи информации с шины данных ЦПЭ) поступают на вход блока формирования управляющих сигналов, а сигнал ПЗХ (подтверждение захвата шин) — на разъем устройства.

Использование в качестве *генератора тактовых импульсов* специальной БИС типа К580ГФ24 либо реализация его на основе микросхем средней и малой степеней интеграции ТТЛ-структуры [1, 2] не уменьшают потребляемой мощности. Поэтому был разработан генератор тактовых импульсов на КМДП-микросхемах серии К564 (рис. 2).

Генератор тактовых импульсов выполнен на микросхемах D2...D5. На элементах D2.1 и D2.2 собран задающий генератор, на элементах D4.1 и D4.2 — делитель частоты на 4, на элементах D2.3 и D3.3 — формирователь длительности импульсов последовательности Ф1, на элементах D2.4 и D3.1 — формирователь временны́х параметров последовательности Ф1, а элементы D3.2 и D3.4 — буферные и дополнительно формируют фронты и срезы импульсов Ф1 и Ф2. Широкий диапазон возможных напряжений питания микросхем серии К564 3...15 В позволил выбрать напряжение питания всех элементов генератора тактовых импульсов равным +12 В. Это дало возмож-

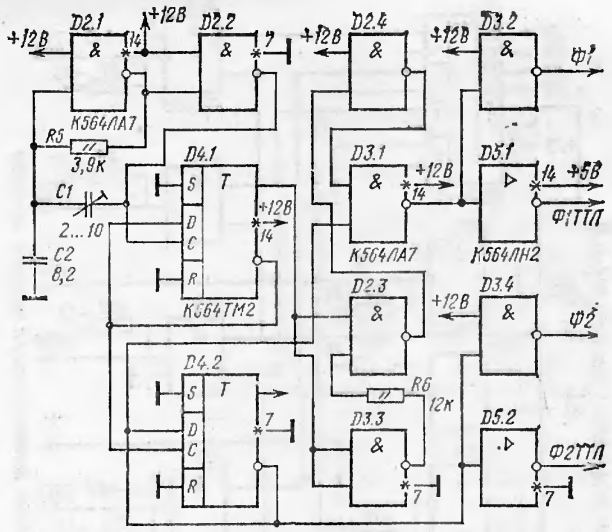


Рис. 2. Принципиальная схема генератора тактовых импульсов

ность без дополнительных затрат получить требуемую для ЦПЭ амплитуду тактовых импульсов Ф1 и Ф2 (+12 В) и обеспечить быстродействие переключения используемых микросхем, лишь незначительно уступающее ТТЛ-схемам. Так, задающий генератор на D2.1 и D2.2 может вырабатывать меандр с частотой 8 МГц и более.

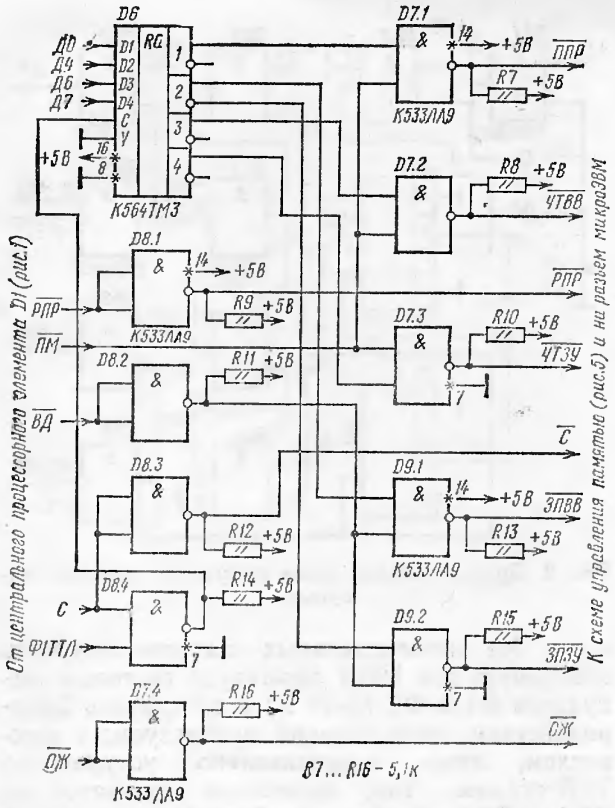
Частоту задающего генератора перестраивает конденсатор C1. Конденсатор C2 повышает устойчивость работы задающего генератора. Длительности фронтов и срезов импульсов Ф1 и Ф2 не превышают 20...25 нс. Это удовлетворяет техническим условиям работы ЦПЭ. При необходимости стабилизировать частоту следования импульсов Ф1 и Ф2 вместо конденсатора C1 включают кварцевый резонатор.

Для получения последовательностей импульсов Ф1 и Ф2 с уровнями ТТЛ используется микросхема К564ЛН2 (элементы D5.1 и D5.2). Она позволяет превышать входным сигналом напряжение питания. В данном случае при амплитуде входных импульсов 12 В и напряжении питания микросхемы 5 В амплитуда импульсов Ф1ТТЛ и Ф2ТТЛ близка к 5 В.

Разработанный генератор тактовых импульсов при частоте следования импульсов Ф1 и Ф2, равной 2 МГц, потребляет 0,16 Вт.

Функцию *системного контроллера* в описываемом устройстве выполняет блок формирования управляющих сигналов.

Выходные сигналы *блока формирования управляющих сигналов* (рис. 3) служат для управления памятью и внешними устройствами, поэтому наряду с малой потребляемой мощностью этот блок должен обладать достаточным быстродействием и нагрузочной способ-


D6
D7.1
D7.2
D8.1
D7.3
D8.2
D8.3
D9.1
D8.4
D9.2
D7.4
К564ТМ3
К533ЛА9
К533ЛА9
К533ЛА9

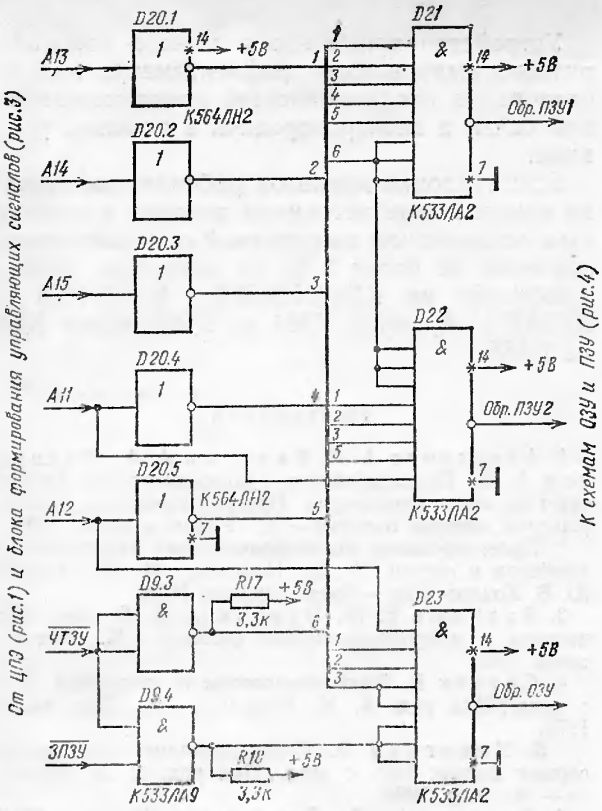


Рис. 5. Принципиальная схема управления памятью

+5 В и время выборки адреса 0,4 мкс. Это хорошо согласуется с быстродействием ЦПЭ. ОЗУ устройства имеет исключительно низкую потребляемую мощность (в режиме обращения не более 200 мВт и значительно меньше в режиме хранения). Для сравнения отметим, что ОЗУ такого же объема, построенное на БИС К541РУ1А и К565РУ2А, потребляет соответственно 3,6 Вт и 13 Вт. Низкая потребляемая мощность и способность микросхем К537РУ2А сохранять записанную информацию при понижении напряжения питания ориентировочно до 1,5 В, при необходимости позволяют легко организовать резервное питание. Сигнал выборки ОЗУ (Обр. ОЗУ) поступает со схемы управления памятью (рис. 5).

Промышленностью выпускается несколько типов микросхем ПЗУ с разнообразными характеристиками. Наиболее оптимально для рассматриваемого устройства — выполнить ПЗУ на основе перепрограммируемых микросхем с ультрафиолетовым стиранием информации типа К573РФ2. Указанные БИС ППЗУ (2К×8 бит) выдерживают жесткие условия эксплуатации, работают от одного источника питания +5 В и имеют режим снижения (ориентировочно на 70 %) потребления мощности при невыборке кристалла по сравнению с режимом считывания информации. ПЗУ устройства (4К×8 бит) выполнено на двух микросхемах типа К573РФ2 D18 и D19 (см. рис. 4). При чтении информации из ППЗУ центральный процессорный элемент работает одновременно лишь с одной БИС, вторая в это время находится в режиме хранения информации. Поэтому максимальная мощность, потребляемая ППЗУ устройства, ориентировочно в 1,3 раза больше мощности потребления одной БИС в режиме обращения и составляет 0,39 Вт. Микросхемы размещены на панельках типа РС-24—8. Это позволяет оперативно заменять их при отладке ПО. Сигналы выборки микросхем ПЗУ (Обр. ПЗУ1, Обр. ПЗУ2) поступают со схемы управления памятью.

Схема управления памятью (см. рис. 5) — это дешифратор адресов микропроцессора A11...A15 и сигналов управления $\overline{\text{ЧТЗУ}}$ и $\overline{\text{ЗПЗУ}}$. Вырабатываемые схемой управления сигналы Обр. ОЗУ, Обр. ПЗУ1 и Обр. ПЗУ2 поступают соответственно на микросхемы ОЗУ и ПЗУ и выбирают поля адресов от 0000 до 0FFF (для ПЗУ) и от 1000 до IFFF (для ОЗУ).

Объемов ОЗУ и ПЗУ по 4 Кбайт достаточно для реализации многих функций малогабаритных переносных приборов, работающих под управлением встроенного устройства. Данные БИС ОЗУ и ПЗУ представляют собой для ЦПЭ по шине адресов нагрузку, не превышающую допустимую (как по току, так и по емкости). Для подключения к шине адресов устройств ввода-вывода и дополнительной памяти использованы неинвертирующие буферные усилители типа К564ПУ4, входящие в буфер шины адреса (см. рис. 1).

К каждому разряду шины данных ЦПЭ подключены лишь по четыре вывода микросхем памяти (гораздо ниже допускаемой нагрузки как по току, так и по емкости). Поэтому устройства ввода и вывода или дополнительную память можно подключать и без применения буферных усилителей данных.

Блоки ввода и вывода для работы с устройством наиболее целесообразно построить на микросхемах КМДП-структуры: регистрах К564ИР2, К564ИР6, К564ИР11, К564ИР12 [6], аналого-цифровых и цифро-аналоговых преобразователях К572ПВ1А, К572ПА1, К572ПА2 [6, 7] с использованием операционных усилителей К140УД12 и К140УД14 [8] с микромощным потреблением.

Разработанное устройство размещается на одной печатной плате размером 220×90 мм из фольгированного стеклотекстолита толщиной 2 мм с двусторонним расположением печатных проводников и односторонним расположением элементов.

Все сигналы шин адресов, данных и управления выведены на разъем, выполненный в ви-

де контактных площадок под розетку типа СНП-112/170×10Р-19—2-В.

Изготовлен макет разработанного устройства.

Основные технические характеристики устройства

Тип центрального процессорного элемента	К580ВМ80
Частота тактовых импульсов, МГц	(1 ... 2)
Разрядность шины данных, бит	8
Объем ПЗУ (с ультрафиолетовым стиранием), бит	4К × 8
Объем ОЗУ, бит	4К × 8
Потребляемая мощность, Вт, не более	1,5
Диапазон рабочих температур, °С	—40 ... + 60
Относительная влажность, %, при 98 °С	35
Конструкция	Одноплатная

Для сравнения энергоемкостей разработанного устройства (≤1,5 Вт) и микроЭВМ, построенной на основе стандартных схемных решений и включающей в себя БИС ЦПЭ — К580ВМ80, ТГ — К580ГФ24, СК — К580ВК28, а также ОЗУ — К541РУ1А (К565РУ2А), ПЗУ — К556РТ5, буферы шины адреса К580ИР83 (К589АП16), укажем, что ее потребляемая мощность (при прочих равных условиях) составляет 12,2...21,8 Вт.

Таким образом, разработанное устройство значительно отличается от существующих (на микропроцессорном комплекте серии КР580 [1—5]) малыми потребляемой мощностью и габаритами, работоспособностью в жестких климатических условиях и может быть встроено в малогабаритные автоматизированные приборы и оборудование.

Устройство применено в макете малогабаритного импульсного рефлектометра для исследования неоднородностей протяженных линий связи и электропередачи в полевых условиях.

Если широкий диапазон рабочих температур не нужен и есть источники питания с возможным отклонением напряжений от номинальных значений не более 5 %, то возможна замена: К580ВМ80 на КР580ИК80А, К537РУ2А на КР537РУ2А, серии К564 на К561, серии К533 на К555.

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах: Программирование, типовые решения, методы отладки.— М.: Радио и связь, 1984.

2. Проектирование микропроцессорных измерительных приборов и систем / В. Д. Циделко, Н. В. Нагаец, Ю. В. Хохлов и др.— Киев: Техника, 1984.

3. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы.— М.: Радио и связь, 1981.

4. Соучек Б. Микропроцессоры и микроЭВМ: Пер. с англ. / Под ред. А. И. Петренко.— М.: Сов. радио, 1979.

5. Клингман Э. Проектирование микропроцессорных систем: Пер. с англ. / Под ред. С. Д. Пашкеева.— М.: Мир, 1980.

6. Ланцов А. Л., Зворыкин Л. Н., Осипов И. Ф. Цифровые устройства на комплементарных МПД-интегральных микросхемах.— М.: Радио и связь, 1982.

7. Шило В. Л. Функциональные аналоговые интегральные микросхемы.— М.: Радио и связь, 1982.

8. Алексенко А. Г., Коломбет Е. А., Стародуб Г. И. Применение прецизионных аналоговых ИС.— М.: Сов. радио, 1980.

Статья поступила 5 февраля 1985 г.

БИНТИ № 18 (2211)

ЭВМ в английских школах

В Англии и Уэльсе в начальных и средних школах для целей обучения используются 120 тыс. микроЭВМ.

Наибольший интерес уроки с применением ЭВМ вызывают у школьников в возрасте 11—12 лет. Чаще всего ЭВМ используются в преподавании таких предметов, как математика и география, и при изучении самой вычислительной техники. Это способствует развитию у детей абстрактного мышления. В результате обучения работе на компьютерах у школьников вырабатываются навыки программирования, но получаемых ими знаний будет недостаточно для того, чтобы стать специалистами по вычислительной технике. Поэтому основное внимание в школах следует уделять не столько изучению самой вычислительной техники, считают английские специалисты, сколько более широкому ее применению при изучении других дисциплин. На уроках можно широко использовать цветные графики и устройства вроде синтезаторов речи, световых перьев, роботы и т. п. Школьники могут вступать в диалог с ЭВМ, причем машинные программы можно выбирать с учетом индивидуальных особенностей детей. С помощью специальных программ во время лабораторных занятий можно моделировать опыты, которые нельзя проводить в школьных условиях.

Особую ценность для развития способностей детей представляют программы поиска информации, которые хранят данные, собранные школьниками. Применение таких программ дает школьникам возможность анализировать собранные ими данные и оценивать их значимость. В начальных классах при обучении письму применяются программы обработки слов.

Традиционные машинные учебные программы предусматривают разделение темы на небольшие разделы, и в конце таких разделов ученику задается вопрос, на который может быть дан только один правильный ответ, и если ученик его находит, то он может переходить к следующему разделу, а при неверном ответе повторяет прежний материал. Такой подход к обучению ограничивает возможности учащихся строго заданными рамками, мешает им разобраться во взаимосвязях изучаемого явления и не способствует развитию их творческих способностей. И его нельзя применять в случаях, когда на заданный вопрос невозможно дать только один ответ.

«New Scientist» (Англия), том 105, № 1449, 1985, с. 34.

В первых двух статьях этого раздела: «Принципы организации и работы дисплеев на основе БИС КР580ВГ75» и «Дисплей для бытовой персональной ЭВМ» на примерах схем алфавитно-цифровых дисплеев с простейшими графическими возможностями рассматриваются принципы организации, режимы работы, особенности программирования и взаимодействия отдельных функциональных блоков дисплея, построенного на трех программируемых БИС: контроллере ЭЛТ (КЭЛТ) — КР580ВГ75, контроллере прямого доступа к памяти (КПДП) — КР580ВТ57 и таймере КР580ВИ53.

УДК 681.327.8.06

Д. И. Панфилов, О. А. Романенко, В. С. Сафанюк, С. Г. Шаронин

ПРИНЦИПЫ ОРГАНИЗАЦИИ И РАБОТЫ ДИСПЛЕЕВ НА ОСНОВЕ БИС КР580ВГ75

Телевизионный растр и видеосигнал

Для дисплеев удобно применять *построчное* отображение информации, в отличие от *чересстрочного*, применяемого в телевидении. Это позволяет упростить схему, формирующую растр, и отказаться от сигналов уравновешивания во время обратного хода кадровой развертки. В телевизионной системе, действующей в СССР (ГОСТ 19432—74), частота строк выбрана равной 15625 Гц, а частота смены кадров при построчном отображении информации — 50 Гц. Таким образом, время отображения одной строки составляет 64 мкс, кадра — 20 мс, а кадр состоит из 312 строк.

Часть телевизионной строки и кадра, которая не используется для отображения информации, служит для задания необходимых границ изображения на экране и гашения луча при обратном ходе строчной и кадровой разверток. На это время выдаются строчные и кадровые сигналы гашения. Длительность отображаемой на экране части строки составляет менее 48 мкс, а кадр при задании приемлемых границ изображения может содержать около 290 видимых строк.

Для синхронизации положения строк и кадров служат строчные и кадровые синхроимпульсы (ССИ и КСИ), определяющие начало обратного хода строк и кадров. ССИ и КСИ формируются во время выдачи строчных и кадровых сигналов гашения и имеют длительность 4 мкс и 180 мкс соответственно.

Упрощенный вид телевизионного видеосигнала для отображения одного кадра информации показан на рис. 1,*а*. Положение кадра на экране задается расположением синхроимпульсов относительно начала строчных и кадровых сигналов гашения. Изменением длительности и положения гасящих сигналов можно добиться расположения отображаемого участка практически в любом месте телевизионного экрана (рис. 1,*б*).

Полоса пропускания видеоусилителя телевизионного приемника достигает 6 МГц, что позволяет выдавать точки в строке с частотой до 12 МГц (один период максимальной частоты видеосигнала содержит две точки). При тактовой частоте 10 МГц (период 100 нс), выбранной в описанной ниже схеме дисплея, можно сформировать на видимом участке строки 48 мкс/100 нс=480 точек.

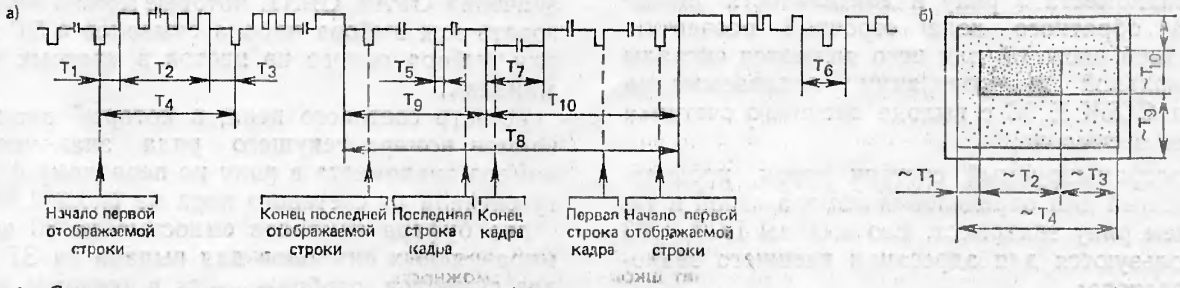


Рис. 1. Структура телевизионного видеосигнала (*а*), расположение видимой части растра на экране (*б*)

Если матрица знакоместа состоит из 6×8 точек, то в одном ряду укладывается максимум 480/6=80 знакомест. Для разделения рядов знакомест необходимо предусмотреть несколько пустых строк, например, три. Так как один отображаемый ряд знакомест по вертикали будет составлять 8 строк, то на экране можно отобразить $280/(3+8)\approx 25$ рядов. Необходимо помнить, что число знакомест в ряду и число рядов ограничено производительностью канала ПДП, а также степенью снижения производительности микроЭВМ. При выборе формата экрана для устранения биений изображений от частоты питающей сети и совмещения цветов при использовании цветного телевизионного приемника рекомендуется использовать стандартные частоты строк и кадров, принятые в ГОСТ.

Таким образом, для формирования изображения на экране электронно-лучевой трубки необходимо, чтобы контроллер ЭЛТ получал из ОЗУ буфера экрана коды очередного ряда отображаемых символов, обеспечивал выдачу их в нужной позиции экрана для дешифрации с помощью знакогенератора (ЗГ), формировал сигналы обратного хода кадровой и строчной разверток и сигналы управления видеосигналом. Все эти возможности реализованы в БИС КР580ВГ75, которая, кроме этого, позволяет формировать курсор в программируемом месте экрана, обрабатывать сигнал от светового пера и отображать простейшие графические символы без использования знакогенератора.

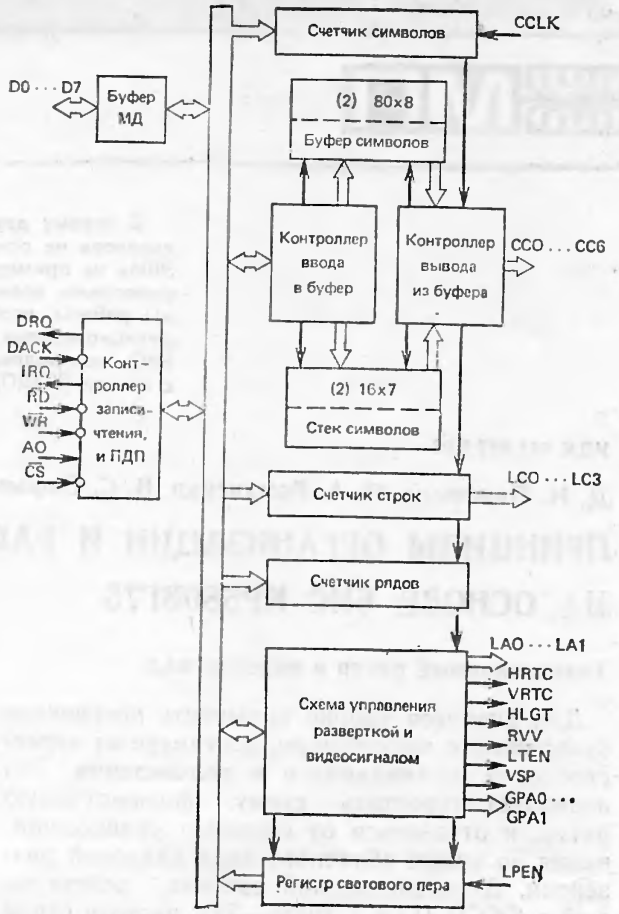


Рис. 2. Структурная схема БИС КР580ВГ75

Структура БИС КР580ВГ75

БИС КР580ВГ75 включает в себя (рис. 2):

двунаправленный трехстабильный буфер для подключения к магистрали данных системы. Буфер содержит регистры команды, параметров и слова состояния, которые служат для управления работой БИС и определения ее текущего состояния;

контроллер чтения, записи и прямого доступа к памяти для управления передачей данных;

программируемый счетчик символов, используемый для определения номера отображаемого знакоместа в ряду и длительности интервала обратного хода строчной развертки. Входной частотой для него являются сигналы символьной синхронизации, подаваемые на вход CCLK БИС с выхода внешнего счетчика точек в символе;

программируемый счетчик строк, предназначенный для определения номера строк в текущем ряду знакомест. Его выходы LC0...LC3 используются для адресации внешнего знакогенератора;

программируемый счетчик рядов, определяющий число рядов знакомест и длительность интервала обратного хода кадровой развертки;

схему управления разверткой и видеосигналом, формирующую сигналы гашения обратного хода строчной и кадровой разверток HRTC и VRTC, сигналы, позволяющие формировать простейшие псевдографические символы LA0, LA1, сигналы, с использованием которых можно реализовать подсветку (увеличенную яркость) — HLGT, мерцание и гашение — VSP, подчеркивание — LTEN, видеореверс символов — RVV, и сигналы общего назначения GPA0, GPA1, которые можно использовать для выбора набора символов в ЗГ или для выбора одного из цветов в цветных терминалах;

регистр светового пера, в который записываются номера текущего ряда знакомест и номера знакоместа в ряду по переднему фронту сигнала от светового пера на входе LPEN;

два буфера символов емкостью по 80 восьмиразрядных символов для выдачи на ЗГ кодов символов, отображаемых в текущем ряду

знакомест и для приема из памяти микроЭВМ по каналу ПДП кодов символов для следующего ряда знакомест;

два стека емкостью по 16 семиразрядных символов для удлинения символьных буферов при использовании режима невидимых управляющих кодов;

контроллеры ввода и вывода, управляющие записью кодов в символьные буферы или стек и выдачей их на ЗГ через выходы СС0...СС6.

Описание принципиальной схемы дисплея

В рассматриваемом дисплее БИС КР580ВТ75 используется для формирования растра, отображения курсора псевдографической алфавитно-цифровой информации и обработки сигнала светового пера; БИС КР580ВТ57 — для передачи данных из буфера экрана во внутренние буферы символов КЭЛТ; БИС КР580ВИ53 — для формирования ССИ и КСИ.

Электрическая принципиальная схема дисплея приведена на рис. 3. Дисплей рассчитан на сопряжение с помощью разъема X1 с микроЭВМ, построенными на основе МПК серии К580 [3, 4]. Разъем X2 служит для подключения светового пера и телевизионного приемника. Память дисплея содержит ОЗУ емкостью 4К байт (D11, D16) и ПЗУ емкостью 4К байт (D12, D13), которые могут быть использованы для организации буфера экрана и хранения программ обслуживания программируемых БИС, ввода-вывода информации и построения псевдографических изображений.

Микросхема D1 является буфером магистрали управления и позволяет отключать на время циклов ПДП сигналы дисплея $\overline{I/OR}$ и $\overline{I/OW}$, предотвращая их ложное срабатывание. Микросхема D3 служит буфером магистрали данных. Дешифратор адреса построен на ПЗУ D8, что дает возможность при необходимости отключать ПЗУ или ОЗУ дисплея.

Схема формирования телевизионного растра состоит из БИС КЭЛТ (D10); знакогенератора (D14); ПЗУ формирования псевдографических символов (D15); генератора тактовой частоты точек и символов (D4.2...D4.5, D18);

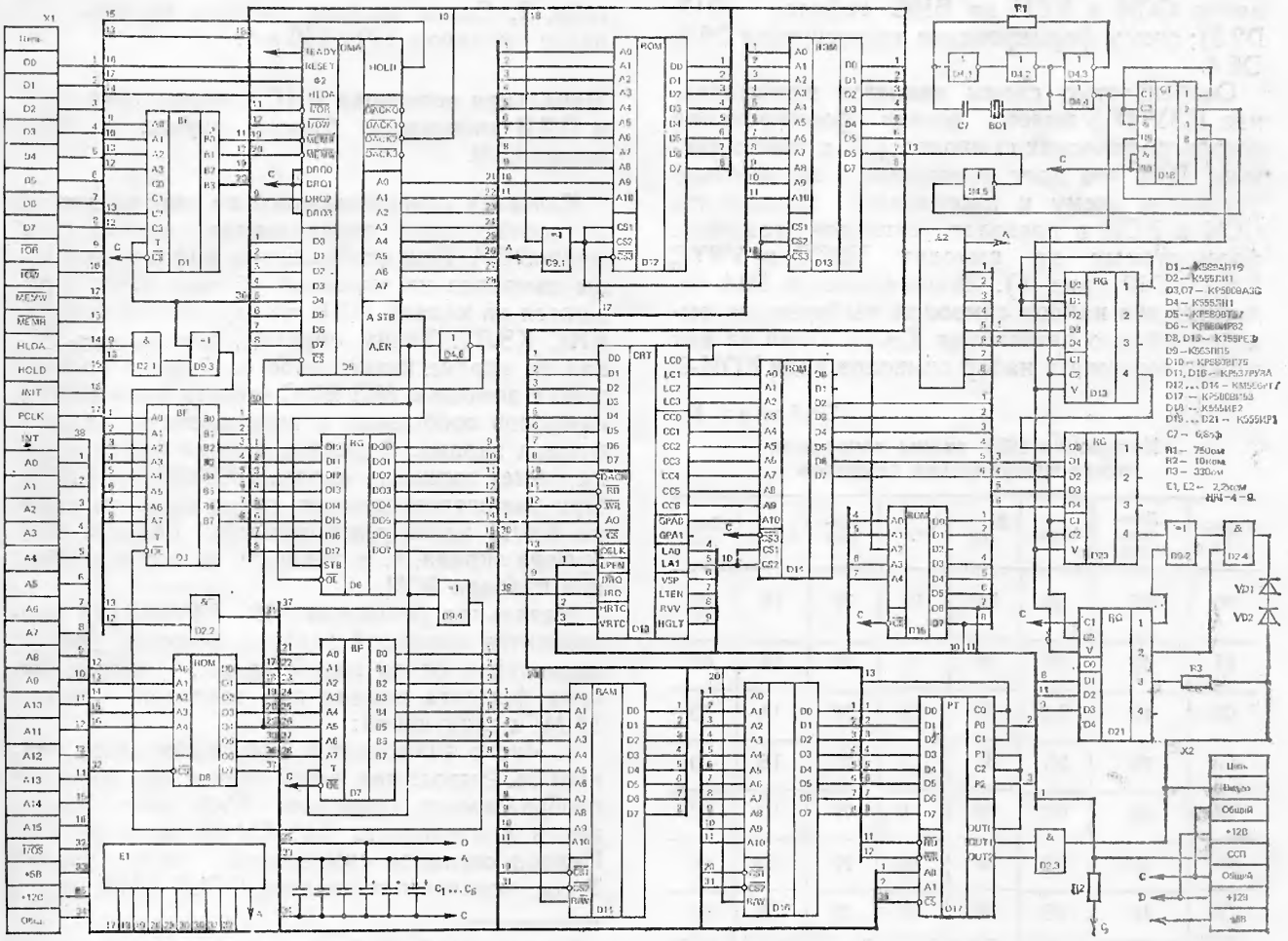


Рис. 3. Электрическая принципиальная схема дисплея

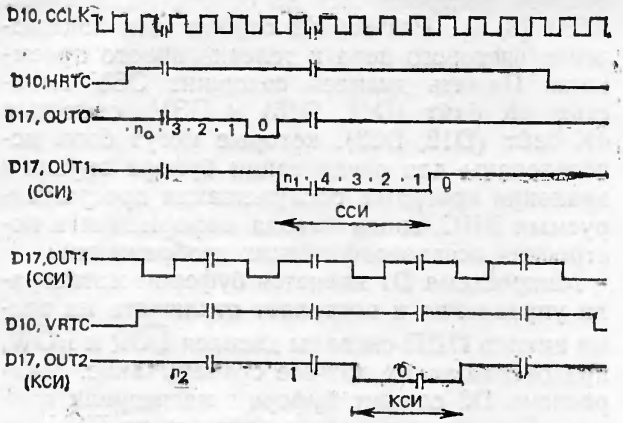


Рис. 4. Временные диаграммы формирования импульсов ССИ и КСИ с помощью таймера

сдвигового регистра для преобразования кода знакогенератора из параллельного в последовательный (D19, D20); регистра для синхронизации сигналов управления видеосигнала, выдаваемых КЭЛТ (D21); схемы формирования ССИ и КСИ на БИС таймера (D17, D2.3); схемы формирования видеосигнала D9.2, D2.4.

Особенностью схемы является использование ПЗУ D15 вместо логики формирования псевдографических символов [2], а также таймера D16, что дает возможность значительно упростить схему и программно перемещать ССИ и КСИ в пределах сигналов гашения, формируемых на выходах HRTC и VRTC КЭЛТ D10 (рис. 4). Знакогенератор D14 содержит два набора символов, выбираемых выходом общего назначения КЭЛТ. Один из них может содержать набор символов кода КОИ-7,

Таблица 1

Кодировка ПЗУ схемы генерации псевдографических символов

Адрес, HEX	Данные, HEX	Адрес	Данные	Адрес	Данные	Адрес	Данные
00	FF	08	BF	10	7F	18	00
01	40	09	3F	11	7F	19	00
02	47	0A	3F	12	7F	1A	00
03	78	0B	3F	13	7F	1B	00
04	BF	0C	BF	14	7F	1C	00
05	40	0D	3F	15	7F	1D	00
06	47	0E	3F	16	7F	1E	00
07	78	0F	3F	17	7F	1F	00

Таблица 2

Адреса внешних устройств и памяти дисплея

Устройство	Метка	Адрес
КПДП	PORT 57— — PORT 57+8	F0 — F8
КЭЛТ	PORT 75— — PORT 75+1	FA — FB
Таймер	PORT 53— PORT 53+3	FC — FF
ПЗУ	—	F000 — FFFF
ОЗУ	—	E000 — EFFF

а второй — набор специальных или псевдографических символов [5]. Кодировка ПЗУ D15 приведена в табл. 1, адреса, присвоенные памяти и интерфейсным БИС дисплея — в табл. 2. Схема дисплея собрана на печатной плате размером 120×240 мм.

Начальная установка БИС контроллеров ЭЛТ и ПДП дисплея и буфера экрана в ОЗУ микроЭВМ

Каждому отображаемому на экране символу соответствует определенная ячейка ОЗУ микроЭВМ. При отображении информации коды символов из экранной области ОЗУ передаются по каналу ПДП во внутренние буферы БИС КЭЛТ. Таким образом, для формирования на экране какого-либо сообщения необходимо с помощью МП БИС занести коды КОИ-7 символов сообщения в определенную область буфера экрана. В данном случае буфер экрана может занимать адреса OE800H...OEFFFH. При включении питания, перед началом работы КЭЛТ необходимо очистить область ОЗУ буфера экрана, т. е. записать во все ее ячейки код пробела 2OH.

Начальная установка БИС КР580ВГ75 производится командой задания формата экрана, имеющей 4 байта параметров *. Порядок выбора формата экрана при тактовой частоте 10 МГц следующий:

1. Число знакомест в ряду выбирается равным 64. Выводимая часть строки, при матрице отображаемых символов 6×8 точек, будет иметь длительность $64 \times 6 \times 100$ нс ≈ 40 мкс. Период сигналов символьной синхронизации T_{CCLK}, подаваемых на вход CCLK БИС, будет

* Зеленко Г. В. Дисплей для бытовой персональной ЭВМ на основе контроллера ЭЛТ КР580ВГ75 — см. следующую статью в этом номере.

равен 6×100=600 нс (минимальный допустимый период для БИС КР580ВГ75 составляет 320 нс).

2. Для разделения рядов предусмотрены четыре пустые строки. При отображении 24 рядов знакомест без пробелов кадр будет содержать 24×(4+8)=288 видимых строк и 312—288=24 невидимых.

3. Строчный сигнал гашения должен иметь длительность 64 мкс — 40 мкг=24 мкс, что составляет 40 периодов частоты T_{CCLK}. Длительность сигнала гашения выбирается равной 32 периодам T_{CCLK}.

4. Кадровый сигнал гашения должен занимать время, необходимое для отображения 24/(8+4)=2 рядов знакомест.

Для формирования ССИ и КСИ используется БИС КР580ВИ53 (см. рис. 3, 4). Нулевой и второй каналы таймера определяют положение ССИ и КСИ относительно строчного и кадрового гасящих сигналов. Первый канал таймера формирует необходимую длительность ССИ. Длительность КСИ постоянна и равна периоду строчной частоты. Необходимо помнить о важности синхронной работы контроллеров ЭЛТ и ПДП, особенно, если не используются прерывания, формируемые БИС КР580ВГ75. Ошибки синхронизации могут привести к смещению информации, записанной в буфере экрана, на экране дисплея.

Возможны несколько способов синхронизации:

с помощью сигнала запроса прерывания, формируемого БИС КЭЛТ на выходе IRQ в конце отображения каждого экрана;

с использованием бита 5 слова состояния КЭЛТ (запрос прерывания), устанавливаемого синхронно с сигналом на выходе IRQ;

с применением команды предустановки счетчиков КЭЛТ;

с использованием искусственного приема, состоящего в том, что загрузка и разрешение работы КПДП производится после загрузки, разрешения работы КЭЛТ и установки им бита 1 слова состояния (недогрузка по ПДП). При этом КЭЛТ начнет запрашивать данные лишь с момента начала отображения следующего кадра и синхронизация будет надежно обеспечена.

ЗАДАНИЕ 1. *Начальная установка дисплея. Временные диаграммы видеосигнала.*

При выполнении задания проводится подготовка буфера экрана в ОЗУ, программирование БИС КР580ВГ75, КР580ВТ57, КР580ВИ53, исследуются временные диаграммы видеосигнала.

1. Заполните ячейки ОЗУ буфера кодом символа КОИ-7, например кодом точки 2EH, для чего введите в память микроЭВМ и выполните программу подготовки буфера LDBUF (программа 1), предварительно записав в аккумулятор МП БИС код выбранного символа.

```
;
; ПРОГРАММА 1
;
LDBUF:  LXI     H,0E800H  ; НАЧ АДРЕС БУФЕРА ЭКРАНА
BUF:    MVI     A,CODE    ; ЗАПИСЬ КОДА СИМВОЛА
        MOV     M,A       ; В ОЗУ
        INX     H
        MOV     A,H       ; ПРОВЕРКА НА КОНЕЦ
        CPI     0F0H      ;
        JNZ     BUF       ;
        RST1              ; СТОП
```

2. Осуществите начальную установку БИС КР580ВГ75, КР580ВТ57 и КР580ВИ53 со следующими парамстрами:

64 знакоместа в ряду, 32 знакоместа обратного хода строчной развертки, 24 ряда знакомест, 2 ряда знакомест обратного хода кадровой развертки, 12 строк в ряду, номер строки подчеркивания — 10, видимый режим кодов признаков, курсор — мерцающий негативный видеоблок, задержка ССИ относительно начала сигнала гашения — 19, длительность сигнала ССИ — 6, задержка КСИ относительно начала сигнала гашения — 11, экранная область занимает адреса 0E800H...0EDFFH ОЗУ.

Для этого введите в память микроЭВМ и выполните программу LDALL (программа 2), предварительно записав в ячейку ОЗУ, соответствующую крайнему левому знакоместу в верхнем ряду, код другого символа, например 30H.

3. Исследуйте временные диаграммы формирования основных составляющих видеосигнала. Для этого, син-

```
;
; ПРОГРАММА 2
;
LDALL:  CALL    TIMER     ; ЗАГРУЗКА КР580ВИ53
        CALL    COMFOR    ;  "   "   КР580ВГ75
        MVI     A,27H     ; РАЗРЕШЕНИЕ ОТОБРАЖЕНИЯ
        OUT     PORT75+1  ;
LD:     IN      PORT75+1  ;
        ANI     02        ;
        JZ      LD        ; ОЖИДАНИЕ НЕДОГРУЗКИ ПДП
        CALL    LDMA      ; ЗАГРУЗКА И ПУСК КР580ВТ57
        MVI     A,84H     ;
        OUT     PORT57+8  ;
        MVI     A,80H     ;   КОМАНДА ЗАПИСИ КУРСОРА
        OUT     PORT75+1  ;
        MVI     A,20H     ;
        OUT     PORT75    ;  НОМЕР РЯДА
        OUT     PORT75    ;  НОМЕР СИМВОЛА
        IN      PORT75+1  ; ПРИ ПУСКЕ СЧИТАТЬ
        OUT     30H       ; СТАТУС КЭЛТ В ВЫХОДНОЙ РЕГ.
        JMP     STAT      ; И ОСТАНОВИТЬСЯ
;
; ЗАГРУЗКА КПДП
;
COMFOR: XRA     A         ; НУ КЭЛТ
        OUT     PORT75+1  ;
        MVI     A,3FH     ; 64 ЗНАКОМЕСТА
        OUT     PORT75    ; В РЯДУ
        MVI     A,57H     ; 24 РЯДА+2РЯДА ОБР.
        OUT     PORT75    ; ХОДА КАДРА
        MVI     A,96H     ; 12 СТРОК В РЯДУ И
        OUT     PORT75    ; ЛИНИЯ ПОДЧ.-10
        MVI     A,4FH     ; 32 ЗНАКОМЕСТА ОБР. ХОДА
        OUT     PORT75    ; СТРОК КУРСОР
        RET               ;
;
; ЗАГРУЗКА ФОРМАТА ЭКРАНА ЭЛТ
;
LDMA:   MVI     A,80H     ; УСТАНОВКА БИТА АВТОЗАГРУЗКИ
        OUT     PORT57+8  ;
        MVI     A,0       ; АДРЕС БУФЕРА ЭКРАНА
        OUT     PORT57+4  ;
        MVI     A,0E8H    ;
        OUT     PORT57+4  ;
        MVI     A,0FFH    ; ДЛИНА БУФЕРА
        OUT     PORT57+5  ;
        MVI     A,85H     ; И РЕЖИМ ПЕРЕДАЧИ
        OUT     PORT57+5  ;
        RET               ;
;
; ЗАГРУЗКА ТАЙМЕРА
;
TIMER:  MVI     A,1AH     ; КАНАЛ 0,РЕЖИМ 5
        OUT     PORT53+3  ;
        MVI     A,52H     ; КАНАЛ 1,РЕЖИМ 1
        OUT     PORT53+3  ;
        MVI     A,9AH     ; КАНАЛ 2,РЕЖИМ 5
        OUT     PORT53+3  ;
        MVI     A,13H     ; ЗАДЕРЖКА ССИ
        OUT     PORT53    ;
        MVI     A,06      ; ДЛИТЕЛЬНОСТЬ ССИ
        OUT     PORT53+1  ;
        MVI     A,0BH     ; ЗАДЕРЖКА КСИ
        OUT     PORT53+2  ;
        RET               ;
```

хронизируя осциллограф сигналом VRTC, зарисуйте осциллограмму видеосигнала на выходе «Видео» дисплея. Укажите на ней сигналы ССИ, КСИ, гасящие кадровые и строчные сигналы, сигнал прямого хода строчной развертки.

ЗАДАНИЕ 2. *Исследование перемещения видимой части растра по экрану и временных диаграмм формирования сигналов синхронизации и гашения.*

Иллюстрируется влияние длительности задержки ССИ и КСИ относительно начала формирования гасящих сигналов на положение видимой части растра на экране, изучаются временные диаграммы работы БИС КЭЛТ и таймера.

1. Осуществите начальную установку дисплея программой LDALL.

2. Подайте сигнал VRTC БИС КЭЛТ на вход синхронизации осциллографа. Исследуйте и зарисуйте взаимное расположение сигналов гашения VRTC КЭЛТ и КСИ, формируемого на выходе OUT2 БИС таймера. Зарисуйте взаимное расположение сигнала HRTC КЭЛТ и ССИ с выхода OUT1 таймера.

3. Измените подпрограмму TIMER для загрузки канала 0 числом 4, а канала 2 числом 3. Убедитесь, что после повторного выполнения программы LDALL видимая часть растра переместится к правому нижнему углу экрана.

4. Проделайте п. 2 при новом расположении ССИ и КСИ.

5. Подберите экспериментально длительность задержки в каналах 0 и 2 для получения центрального расположения растра и запишите полученные значения в подпрограмму TIMER.

ЗАДАНИЕ 3. *Задание формата экрана.*

Иллюстрируется возможность изменения формата экрана в пределах устойчивой синхронизации строчной и кадровой разверток телевизионного приемника с помощью команды задания формата экрана БИС КЭЛТ.

1. Измените подпрограммы загрузки КЭЛТ (COMFOR) и КПДП (LDMA) для получения формата экрана размером 32 ряда по 64 знакоместа и 16 рядов по 80 знакомест.

2. Выполните измененные программы, сравните число отображаемых символов с заданным, измерьте при помощи осциллографа частоту ССИ, КСИ, длительность ССИ, КСИ гасящих сигналов.

Вывод информации на дисплей. Коды управления и визуальных признаков КЭЛТ КР580ВГ75

Весь буфер экрана в ОЗУ можно разграничить по рядам знакомест. При этом левой верхней позиции экрана будет соответствовать ячейка ОЗУ с наименьшим адресом. Для вывода символа в заданную позицию экрана необходимо записать его код по адресу: АДРЕС=(НОМЕР РЯДА)×(ЧИСЛО СИМВОЛОВ В РЯДУ)+(НОМЕР В РЯДУ)+(НАЧАЛЬНЫЙ АДРЕС БУФЕРА ЭКРАНА).

Существуют два способа задания псевдографических символов. Первый состоит в том, что выдачей на выходы универсальных признаков (GPA1, GPA0) кода 01 можно отключить ПЗУ ЗГ алфавитно-цифрового набора кодов КОИ-7 и подключить ПЗУ специальных символов. При этом пользователь может поставить в соответствие каждому 7-разрядному коду необходимый специальный графический (например, при создании игровых программ) или псевдографический символы (например, наклонная черта, треугольник и т. п.) Второй способ основан на возможности КЭЛТ генерировать простейшие графические символы [2] без использования ПЗУ ЗГ. В этом случае необходимый символ задается с помощью 8-разрядного кода псевдографического символа. Признаки этих символов позволяют задать мерцание, подсвечивание или их комбинацию. Признаки таких псевдографических символов имеют более высокий приоритет, чем визуальные признаки поля символов, в пределах которого они находятся.

В процессе обработки сигналов с выходов RW и VSP КЭЛТ схемой формирования видеосигнала, псевдографический символ может оказаться в поле негативного изображения, поскольку при его формировании используется сигнал VSP, имеющий различный приоритет относительно сигнала RVV. Это приводит к тому, что часть псевдографического символа, формируемая с помощью сигнала VSP, отображается обычно или негативно. Для последовательности символов, генерируемых ЗГ, можно устанавливать отображение с мерцанием, подсвечиванием, видеореверсом, подчеркиванием или их комбинацией с использованием кодов визуальных признаков.

ЗАДАНИЕ 4. *Вывод текстовых сообщений на экран дисплея. Визуальные признаки. Коды псевдографических символов.*

Демонстрируется процесс формирования сообщений в экранной области, действие на изображение кодов визуальных признаков и возможность построения простейших графических изображений без использования знакогенератора.

1. Исследуйте влияние кодов визуальных признаков на отображение информации на экране. Обратите внимание, что знакоместо кода визуального признака изображается в виде пустого знакоместа, так как был установлен режим видимых признаков.

2. Исследуйте отображение на экране псевдографических символов без участия ЗГ. Изменяя комбинации младших двух битов этих кодов, задайте индивидуальные признаки мерцания и подсветки. Убедитесь, что

```
;
;
; ПРОГРАММА 3
;
RAMK:  LXI    H,0E800H ;
       LXI    D,0E0FFH ;
       MVI    B,40H    ;
       MVI    A,0E0H   ;
RAM1:  MOV    M,A      ; ПРОРИСОВКА ГОР. ЛИНИЙ
       STAX   D        ;
       INX    H        ;
       DCX    D        ;
       DCR    B        ;
       JNZ    RAM1     ;
       MVI    B,18H    ;
       LXI    D,0841H  ;
RAM2:  MOV    M,A      ; ПРОРИСОВКА ВЕРТ. ЛИНИЙ
       DCX    H        ;
       MOV    M,A      ;
       INX    D        ;
       DCR    B        ;
       JNZ    RAM2     ;
       MVI    A,0C0H   ; ПРОРИСОВКА УГЛОВ
       STA    0E800H   ;
       MVI    A,0C4H   ;
       STA    0E83FH   ;
       MVI    A,0C8H   ;
       STA    0E040H   ;
       MVI    A,0CCH   ;
       STA    0E0FFH   ;
       RST1            ; СТОП
```

признаки кодов псевдографических символов, кроме видеореверса, имеют более высокий приоритет, чем визуальные признаки, заданные на область, где они записаны.

3. Используя коды псевдографических символов, выведите на экран рамку, для чего введите в память микроЭВМ и выполните программу RAMKA (программа 3).

ЗАДАНИЕ 5. *Коды управления.*

Иллюстрируется возможность изменения числа знакомест в ряду и рядов в экране под воздействием кодов управления.

1. Заполните с помощью программы LDBUF буфер экрана кодами любого символа.

2. Убедитесь, что часть знакомест в ряду и рядов может быть очищена без записи кода пробела в буфер экрана: после записи в ОЗУ кода окончания ряда (0F0H) гасится часть ряда правее него, а после записи кода окончания кадра (0F2H) — часть знакомест в кадре правее и ниже его. Синхронность работы КЭЛТ и КПДП не нарушается.

3. Проверьте возможность нарушения синхронности работы БИС КР580ВГ75 и КР580ВТ57 при использовании кодов, задающих останов ПДП. Для этого после повторного выполнения пункта 1 записью в предпоследнюю ячейку экранного буфера (0EDFEH) кода 0F3H уменьшите на единицу число символов, передаваемых по каналу ПДП в каждом отображаемом кадре. При этом изображение на экране начнет перемещаться, так как КЭЛТ и КПДП выйдут из синхронизма из-за автозагрузки последнего прежним значением длины экранного буфера. Укажите, как с использованием прерываний можно устранить этот недостаток.

4. Убедитесь, что если специальные коды, задающие останов ПДП, не являются последними символами в пачке или ряду, то КЭЛТ считывает еще один символ, следующий за ними.

ЗАДАНИЕ 6. *Неотображаемые управляющие коды.*

Рассматриваются особенности работы БИС КЭЛТ и КПДП при использовании режима неотображаемых управляющих кодов. Изучается работа внутреннего стека символов.

1. Заполните с помощью программы LDBUF буфер экрана кодом символа пробела 20H.

2. Измените подпрограмму COMFOR для задания режима формирования невидимых признаков поля и выполните программу LDALL.

3. Убедитесь, что запись кода визуального признака увеличивает число символов, запрашиваемых КЭЛТ по ПДП на единицу и синхронность работы КЭЛТ и КПДП нарушается.

4. Восстановите синхронизм работы БИС: для этого запишите в ячейку ОЗУ 0EDFFH код окончания кадра с остановом ПДП (0F3H) и повторно выполните программу LDALL; число запросов ПДП, выдаваемых КЭЛТ, уменьшится на единицу. Обратите внимание, что код визуального признака не занимает знакоместа на экране.

5. Сформируйте ряд символов с использованием неотображаемых управляющих кодов. Для этого запишите в ячейки экранного ОЗУ второго ряда сообщение 'KA KB KC ...KP KQ KR', где K — код визуального признака подчеркивания (0A0H). Так как в одном ряду будет присутствовать 18 кодов, а глубина стека КЭЛТ равна 16, стек переполнится [1], занесенные в него первыми коды букв «A» и «B» будут утеряны, а вместо них появятся буквы «Q» и «R».

6. Убедитесь, что бит переполнения стека в слове состояния КЭЛТ установлен в единицу, считав его в регистр A МП БИС.

7. Проверьте факт потери старшего бита при записи числа в стек КЭЛТ. Для этого замените код буквы «C» во втором ряду кодом признака видеореверса с подчеркиванием (0B0H). При этом из-за потери старшего разряда он будет интерпретироваться как код цифры 0 (30H).

Исследование работы БИС КЭЛТ в различных режимах обмена данными по каналу ПДП

БИС КР580ВГ75 можно запрограммировать командой разрешения отображения на различные режимы формирования запроса на ПДП. Передача данных по каналу осуществляется пакетами, интервал между которыми и число символов в пакете задаются программно.

Если интервал между пакетами равен нулю, то заполнение буферов символов КЭЛТ происходит наиболее быстро, но при этом МП БИС длительное время находится в состоянии ЗАХВАТ и не реагирует на запросы прерывания от внешних устройств. При работе в таком режиме неэффективно используются такты T4 и T5 машинных циклов МП БИС [6], что вызывает увеличение времени выполнения основной программы микроЭВМ.

С другой стороны, при большом интервале между пакетами и малом числе символов в них, может произойти недогрузка буфера символов КЭЛТ, что приведет к запросу отображения информации. В сложных системах недогрузка буфера возникает в том случае, когда ПДП осуществляет канал БИС контроллера ПДП с более высоким приоритетом, чем канал, обслуживающий КЭЛТ.

ЗАДАНИЕ 7. *Временные диаграммы обмена информацией по каналу ПДП КЭЛТ.*

Исследуются временные диаграммы обмена данными по каналу ПДП в различных режимах формирования запроса на ПДП БИС КЭЛТ.

1. Задайте командой разрешения отображения КЭЛТ в программе LDALL несколько различных режимов формирования запроса на ПДП и выполните программу. Оцените, какую часть времени КПДП захватывает магистраль микроЭВМ. Синхронизируя осциллограф задним фронтом сигнала на выходе HRTC КЭЛТ и подключая вход «У» осциллографа к выводам DRQ и DACK КЭЛТ, зарисуйте осциллограммы сигналов на этих выходах.

2. Проанализируйте снижение производительности МП БИС для двух режимов формирования запроса ПДП с помощью программы выдачи звуковых сигналов [3], которая записывает нули и единицы в одноразрядный регистр звукового выхода [3] со звуковой частотой. По снижению частоты звукового сигнала можно определить факт захвата КПДП большего числа циклов доступа к магистрали или неэффективного использования тактов T4 и T5. Для этого, после выполнения программы LDALL осуществите пуск программы MAIN [3], предварительно набрав на переключателях входного регистра код длительности задержки 0B0H.

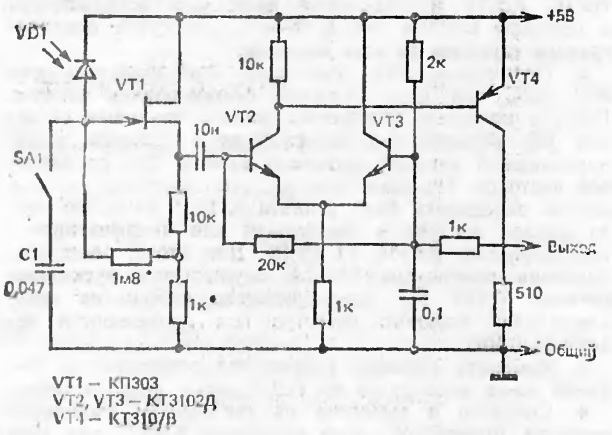
3. Измените команду разрешения отображения так, чтобы из-за недогрузки по ПДП экран дисплея погас.

4. Считайте и выведите на светодиоды выходного регистра микроЭВМ слово состояния КЭЛТ, для чего выполните один раз программу STAT. Убедитесь, что в слове состояния установлен бит 1 слова состояния (недогрузка по ПДП).

Отображение курсора и обработка сигналов светового пера в КЭЛТ

БИС КЭЛТ позволяет формировать курсор в любом знакоместе экрана. Его положение задается записью в соответствующие внутренние регистры КЭЛТ номера знакоместа в ряду и номера ряда. Если номер знакоместа или номер ряда больше, чем указано в команде, определяющей формат экрана, курсор отображаться не будет. При выполнении команды задания экрана можно выбрать один из четырех видов отображения курсора: мерцающее или немерцающее негативное знакоместо, мерцающее или немерцающее знакоместо с подчеркиванием.

Особенностью КЭЛТ является наличие схемы обработки сигнала светового пера. По переднему фронту этого сигнала, возникающего при прохождении электронного луча ЭЛТ около фотодиода, в регистры светового пера заносится текущая информация из счетчика знакомест и счетчика рядов. На наличие факта записи указывает установленный в «1» бит 4 слова состояния КЭЛТ (сигнал светового пера). МП БИС может командой чтения регистров светового пера считать их содержимое и использовать для организации обратной связи от оператора к микроЭВМ. Схема усиления сигнала от фотодатчика приведена на рис. 5. При замкнутой клавише SA1 (световое перо не прижато к экрану) вход схемы зашунтирован конденсатором С1. На выходе схемы уровень «Лог.0». При размыкании клавиши SA1 нажатием на световое перо, схема переходит в режим усиления тока фотодиода. Прохождение электронного луча мимо фотодиода увеличивает в нем обратный ток, что приводит к увеличению потенциала на истоке транзистора VT1. Увеличение его на 0,2 В приводит к перебросу схемы переключателя тока VT2, VT3 и открыванию транзистора VT4, что создает на выходе схемы переход сигнала с низкого на высокий логический уровень.



VT1 – КП303
VT2, VT3 – КТ3102Д
VT4 – КТ3107В

Рис. 5. Электрическая принципиальная схема светового пера

ЗАДАНИЕ 8. *Курсор и его позиционирование на экране.*

Изучаются виды курсора, формируемого КЭЛТ, способ его позиционирования и отображение на полях символов с различными визуальными признаками.

1. Измените в программе LDALL команду загрузки позиции курсора так, чтобы он занимал третье знакоместо в третьем ряду экрана. Заполните буфер экрана каким-нибудь символом с помощью программы LDBUF.

2. Исследуйте отображение всех видов курсора на следующих полях символов: поле видеореверса, поле мерцающего видеореверса, подсвеченное поле с подчеркиванием, мерцающее подсвеченное поле с подчеркиванием.

ЗАДАНИЕ 9. *Световое перо.*

Изучается работа со световым пером (СП) и способ коррекции задержки поступления сигнала от его фотодатчика.

1. С помощью программы LDBUF заполните буфер экрана дисплея кодом какого-либо символа.

2. Определите число знакомест, на которое необходимо скорректировать содержимое регистров СП, пользуясь программой LPCUR (программа 4), которая устанавливает курсор в позицию, определяемую СП.

3. Занесите результат в первую команду программы и проверьте правильность позиционирования.

```
;
; ПРОГРАММА 4
;
LPCUR:  MVI     C,CORR  ; ЧИСЛО ЗНАКОМЕСТ КОРРЕКЦИИ
LPEN:   IN      PORT75+1 ; ОЖИДАНИЕ СИГНАЛА
        ANI     10H     ; СВЕТОВОГО ПЕРА
        JZ      LPEN    ;
        MVI     A,60H   ; ЧТЕНИЕ РЕГИСТРОВ
        OUT     PORT75+1 ; СВЕТОВОГО ПЕРА
        IN      PORT75  ;
        MOV     L,A     ;
        IN      PORT75  ;
        MOV     H,A     ;
        MOV     A,C     ; КОРРЕКЦИЯ
        DCR     A       ;
        SUB     L       ;
        CMA             ;
        JC      LP      ;
        ADI     40H     ;
        DCR     H       ;
LF:     MOV     L,A     ;
        MVI     A,80H   ; КОМАНДА УСТАНОВКИ
        OUT     PORT75+1 ; КУРСОРА
        MOV     A,L     ;
        OUT     PORT75  ;
        MOV     A,H     ;
        OUT     PORT75  ;
        JMP     LPEN    ; ПРОДОЛЖАТЬ
;
;
```

Исследование прерываний, формируемых КЭЛТ

В случае разрешения прерываний после начальной установки, КЭЛТ формирует на выходе IRQ сигнал запроса прерывания в начале отображения последнего ряда знакомест экрана. Для обслуживания этого прерывания МП БИС имеет время, равное запрограммированной длительности импульса на выходе VRTC, так как КЭЛТ начинает заполнение внутреннего буфера символов в начале последнего ряда обратного хода кадровой развертки. Выдача запросов прерываний разрешается командами разрешения прерывания и разрешения отображения КЭЛТ, а запрет осуществляется командами запрещения пре-

рывания и задания формата экрана. О факте разрешения прерывания говорит установленный в «1» бит 6 слова состояния КЭЛТ (Разрешение прерывания). Кроме этого, бит 5 слова состояния (Запрос прерывания) позволяет программно считывать состояние выхода IRQ. Сброс сигнала запроса прерывания осуществляется при выполнении команды задания формата экрана. Следует отметить, что при включении питания можно установить запрос прерывания КЭЛТ. Поэтому до разрешения прерываний МП БИС должна произвести операцию чтения слова состояния КЭЛТ.

Запрос прерывания может быть использован для перегрузки КПДП, что позволяет КЭЛТ надежно синхронизироваться с каналом ПДП и дает возможность осуществлять быстрое перемещение изображения на экране за счет перемещения границ буфера экрана в ОЗУ. Для реинициализации КПДП можно также использовать сигнал с выхода конца счета канала, обслуживающего КЭЛТ.

Для синхронизации с использованием прерываний необходимо загрузить КЭЛТ и разрешить ему прерывания по программе начальной установки системы и загрузить КПДП по программе обслуживания запроса прерывания.

При использовании совместно с КЭЛТ второго канала КПДП в режиме автозагрузки можно осуществлять сдвиг рядов символов на экране на любое число рядов без перезаписи области ОЗУ, содержащей данные дисплея. В качестве примера можно рассмотреть режим работы, часто используемый в алфавитно-цифровых дисплеях, при котором каждый ряд изображения передвигается вверх по мере начала образования нового ряда. При применении автозагрузки каждый адрес ОЗУ привязан к конкретному символу экрана даже при передвижении символа вверх по экрану. Для этого буфер экрана разделен на два блока (рис. 6), первый определяет область

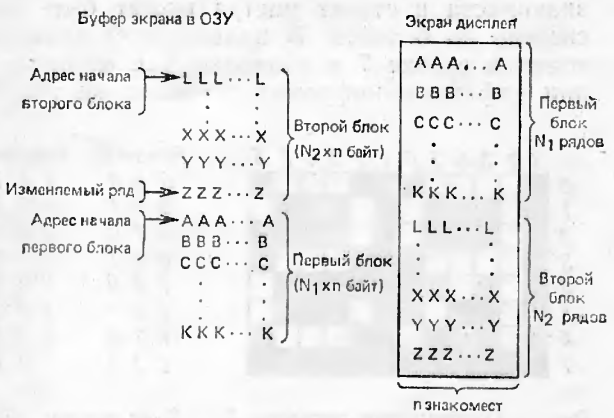


Рис. 6. Организация сдвига изображения на экране

ОЗУ от верхней части экрана до конца буфера ОЗУ, а второй — от начала буфера ОЗУ до нижней части экрана. По мере добавления рядов начальный адрес первого блока и конечный адрес второго изменяются таким образом, чтобы переместить ряды изображения на экране. После полного заполнения экрана каждый новый ряд символов стирает и переписывает старый ряд. Это приводит к необходимости повторного циклического прохождения переменных адресов блоков через начало буфера экрана. Изменение адресов блоков выполняется программой обслуживания прерывания КЭЛТ на основании расположения нового ряда символов в буфере экрана. При этом параметры первого блока выдаются во второй канал ПДП, а параметры второго блока — в третий. После этого КПДП готов к передаче всего массива символов для отображения экрана.

ЗАДАНИЕ 10. *Обслуживание запросов прерывания КЭЛТ.*

Рассмотрены возможность перезагрузки КПДП в конце отображения кадра по запросу прерывания от КЭЛТ для организации сдвига текста на экране и действие команд разрешения-запрещения прерываний.

1. Выполните начальную установку микроЭВМ. Введите в ОЗУ программу начальной установки дисплея LDINT и программу обслуживания запроса прерыва-

```
                ;
                ; ПРОГРАММА 5
                ;
; ФРАГМЕНТ ПРОГРАММЫ НАЧАЛЬНОЙ УСТАНОВКИ СИСТЕМЫ
                ;
LDINT:  CALL    TIMER       ; ЗАГРУЗКА ТАЙМЕРА
        CALL    COMFOR      ; ЗАГРУЗКА КЭЛТ
        IN      PORT75+1    ; РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ
        MVI     A,27H       ; КОМАНДА "СТАРТ ДИСПЛЕЯ"
        OUT     PORT75+1    ;
                ;
        EI                  ; РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ
                            ; СИСТЕМЫ
                ;
                ; ФРАГМЕНТ ПРОГРАММЫ ОБРАБОТКИ ПРЕРЫВАНИЯ ОТ КЭЛТ
                ;
RENOVA: IN      PORT75+1    ; СЧИТЫВАНИЕ СОСТОЯНИЯ
        XRA     A
        MOV     H,A
        OUT     PORT57+3    ; ЗАПРЕТ ПДП
        LXI     D,0E800H    ; НАЧАЛЬНЫЙ АДРЕС ЭКРАННОГО БУФЕРА
        OUT     PORT57+6    ; МЛАДШИЙ АДРЕС НАЧАЛА БУФЕРА
        MOV     A,D         ; СТАРШИЙ АДРЕС
        OUT     PORT57+6    ; В РЕГИСТР КАНАЛА ПДП3
        MVI     A,0FFH      ; МЛАДШИЙ БАЙТ ДЛИНЫ
        OUT     PORT57+7    ; В РЕГИСТР ПДП3
        MVI     A,85H       ; СТАРШИЙ БАЙТ ДЛИНЫ И РЕЖИМ
        IN      20H         ; ЧИСЛО С ВХОДНОГО РЕГИСТРА
        ANI     0FH         ; МАСКА
        MOV     L,A         ;
        DAD     H           ;
        DAD     H           ;   УМНОЖЕНИЕ НА 64
        DAD     H           ;  (КОЛИЧЕСТВО СИМВОЛОВ В РЯДУ )
        DAD     H           ;
        DAD     H           ;
        DAD     H           ;
        XCHG                ; ФОРМИРОВАНИЕ НОВОГО НАЧАЛЬНОГО
        DAD     D           ;  АДРЕСА
        MOV     A,L         ;
        OUT     PORT57+4    ; МЛАДШИЙ БАЙТ НАЧАЛА БУФЕРА
        MOV     A,H         ; СТАРШИЙ БАЙТ
        OUT     PORT57+4    ; В РЕГИСТР АДРЕСА КАНАЛА ПДП3
        MVI     A,0FFH      ; ВЫЧИСЛЕНИЕ МЛАДШЕГО БАЙТА
        SUB     E           ;  ДЛИНЫ И
        OUT     PORT57+5    ; ЗАПИСЬ В РЕГИСТР ДЛИНЫ ПДП3
        MVI     A,85H       ; ВЫЧИСЛЕНИЕ СТАРШЕГО БАЙТА
        SBB     D           ;  ДЛИНЫ И РЕЖИМА
        OUT     PORT57+5    ;  И ЗАПИСЬ В РЕГИСТР ПДП3
        MVI     A,34H       ; УСТАНОВКА БИТА АВТОЗАГРУЗКИ
        OUT     PORT57+8    ; РАЗРЕШЕНИЕ РАБОТЫ ПДП2
                ;
                ;
                ; ИНИЦИАЛИЗАЦИЯ КОНТРОЛЛЕРА ПРЕРЫВАНИЯ
                ;
                ;
        EI                  ; РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ
        RET
        END
```

ния КЭЛТ RENDMA (программа 5). Занесите в таблицу передачи управления [2] ее адрес.

2. Заполните буфер экрана кодом какого-либо символа с помощью программы LDBUF, а затем занесите в ячейки, соответствующие первым двум знакоместам всех рядов, их порядковый двузначный номер.

3. Убедитесь, что в случае разрешения прерываний, изображение на экране смещается на число рядов, определяемое кодом, набранным на переключателях входного регистра микроЭВМ.

4. Выполните программу STAT и проверьте наличие установленного в «1» бита 6.

5. Проверьте, как будет отображаться информация на экране в случае запрещения прерываний.

Предложенная схема дисплея позволяет реализовать все возможности БИС КЭЛТ. Совместно с бытовым телевизионным приемником или промышленными серийными мониторами схему можно использовать для проведения исследований принципов организации, программирования и работы дисплеев на основе БИС КР580ВГ75, КР580ВТ57, КР580ВИ53, а также в составе персональной ЭВМ для отображения информации.

ЛИТЕРАТУРА

1. Горбунов В. Л., Панфилов Д. И., Преснухин Л. Н. Микропроцессоры. Основы построения микроЭВМ.— М.: Высшая школа, 1984.— 144 с.

2. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.— 270 с.

3. Горбунов В. Л., Панфилов Д. И. Микропроцессоры. Лабораторный практикум.— М.: Высшая школа, 1984.—103 с.

4. Преснухин Л. Н., Панфилов Д. И., Романенко О. А., Шаронин С. Г. Лаборатория по изучению микропроцессорных комплектов с фиксированным набором команд.— Микропроцессорные средства и системы. 1985, № 1, с. 77—81.

5. David A. Kater, Susan J. Thomas. TRS-80 Graphics for the model I and III. 1982.

6. Романенко О. А., Сафанюк В. С., Панфилов Д. И., Шаронин С. Г. Устройство прямого доступа к памяти микроЭВМ.— Микропроцессорные средства и системы, 1985, № 2, с. 65—75.

Статья поступила 13 июня 1985 г.

УДК 681.327.13

Г. В. Зеленко

ДИСПЛЕЙ ДЛЯ БЫТОВОЙ ПЕРСОНАЛЬНОЙ ЭВМ

Микросхема КР580ВГ75 является программируемым контроллером растрового алфавитно-цифрового дисплея с электронно-лучевой трубкой (ЭЛТ). Она предназначена для работы в микропроцессорных системах на базе микропроцессора КР580ИК80А, хотя и не исключается ее использование в других системах. Контроллер позволяет сократить число схем, реализующих функции дисплея, и тем самым уменьшить стоимость и повысить надежность разрабатываемой аппаратуры. Микросхема может применяться в сложных специализированных дисплеях, а также в качестве составной части микроЭВМ, формирующей видеосигнал для обычного телевизора, используемого в ней в качестве видеотерминала. Прежде чем перейти к описанию контроллера ЭЛТ, рассмотрим особенности формирования алфавитно-цифровых символов на экране растровых дисплеев.

Формирование символов на экране растровых дисплеев

Символы на экране ЭЛТ формируются путем засветки отдельных точек телевизионного растра и располагаются в фиксированных позициях — знакоместах. Под каждое знакоместо отводится определенное число строк телевизионного растра, а в пределах одного знакоместа на каждой строке в зависимости от отображаемого символа может быть засвечено несколько точек. Максимальное число таких точек, а также число телевизионных строк, отведенных на одно знакоместо, определяют формат отображаемых символов. По горизонтали знакоместа обычно расположены вплотную, одно к другому, образуя ряды знакомест.

Например, для формирования символов на экране можно использовать матрицу точек форматом 8×6 (рис. 1). Это означает, что каждое знакоместо занимает на экране 8 строк телевизионного растра, а в пределах знакоместа в строке растра может быть засвечено до 6 точек. В приведенном примере точки в строке 7 и столбцах 1 в отображении алфавитно-цифровых символов не участ-

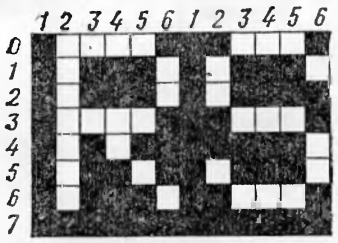


Рис. 1. Отображение символов R и S на экране ЭЛТ растрового дисплея и режимы счета строк растра в знакоместе

вуют, благодаря чему на экране образуются промежутки между символами.

Для формирования символов на экране ЭЛТ в схемах алфавитно-цифровых дисплеев обычно присутствует так называемое ПЗУ знакогенератора. В нем хранится информация о графическом представлении отображаемых символов. Засветка точек (модуляция луча ЭЛТ) в алфавитно-цифровых дисплеях обычно происходит по сигналам с выхода сдвигового регистра, в который предварительно в параллельной форме заносится код из ПЗУ знакогенератора.

Частота, с которой осуществляется сдвиг информации в сдвиговом регистре, определяется числом точек в строке в пределах знакоместа, числом знакомест в ряду и временем развертки одной строки телевизионного растра. Интервал времени, за который луч на экране ЭЛТ проходит через одно знакоместо, определяет период сигнала «Символьная синхронизация».

Применительно к примеру на рис. 1 (т. е. к формату матрицы 8×6) каждому отображаемому символу в ПЗУ знакогенератора соответствует группа из восьми последовательно расположенных ячеек памяти, в которых записана информация о том, какие шесть точек в каждой из восьми строк растра в пределах знакоместа должны засвечиваться при отображении соответствующего символа. Нулевое значение бита в ячейке памяти знакогенератора определяет местоположение засвечиваемой точки на экране в пределах знакоместа. Фрагмент содержимого ПЗУ знакогенератора приведен в табл. 1. Если сигнал с выхода сдвигового регистра проинвертировать, то на экране будут отображаться символы в негативном виде — темные точки на светлом фоне.

Выбор группы ячеек знакогенератора, соответствующей текущему отображаемому символу, определяется кодом последнего. Это 7-разрядный стандартный код КОИ-7 или другой аналогичный, например ASCII. Код символа подается на адресные линии А3 ... А9 ПЗУ знакогенератора. На линии А0 ... А2 подается код с двоичного счетчика номеров строк в пределах знакомест, определяющий, из какой ячейки знакогенератора в выбранной группе будет считан код в сдвиговый регистр. В течение развертки одной строки растра состояние счетчика номеров строк в пределах знакомест остается неизменным, в то время как коды символов на адресных линиях А3 ... А9 ПЗУ знакогенератора могут меняться в зависимости от отображаемых символов при достижении лучом ЭЛТ каждого нового знакоместа. Изменение кодов символов происходит под воздействием сигналов «Символьная синхронизация», период следования которых определяется временем прохождения лучом ЭЛТ одного знакоместа и обозначается как T_{CCLK}.

При развертке следующей строки телевизионного растра код с выхода счетчика увеличивается на 1, а на адресных линиях А3 ... А9 вновь повторяется та же последовательность кодов, что и при развертке предыдущей строки. Этот процесс повторяется, пока не завершится развертка всех строк телевизионного растра для ряда знакомест, после чего счетчик номеров строк растра в пределах знакомест будет установлен в «нулевое» состояние, а на адресные линии А3 ... А9 начнут поступать коды для отображения символов в следующем ряду знакомест. В табл. 1 иллюстрируется состояние адресов групп ячеек ПЗУ знакогенератора, где хранится информация для отображения символов R и S, кодам этих символов. Символы R и S кодируются в КОИ-7 как 1010010 и 1010011 соответственно.

Как правило, в микропроцессорных дисплеях каждому знакоместу на экране ЭЛТ соответствует определенная ячейка ОЗУ в так называемой экранной области ОЗУ. Микропроцессор может работать с экранной областью так же, как и с любой другой областью ОЗУ. Для вывода символа на экран на определенное знакоместо микропроцессор должен записать его код в ячейку экранной области ОЗУ.

Чтобы изображение постоянно присутствовало на экране ЭЛТ, необходимо периодиче-

Таблица 1

Фрагмент содержимого ПЗУ знакогенератора

Адреса ячеек ПЗУ знакогенератора		Коды с выхода ПЗУ	
А	Б		
010100110	000	100001	
010100110	001	101110	
010100110	010	101110	Группа ячеек с информацией о графическом представлении символа
010100110	011	100001	
010100110	100	101011	
010100110	101	101101	
010100110	110	101110	
010100110	111	111111	
010100111	000	110001	
010100111	001	101110	
010100111	010	101111	Группа ячеек с информацией о графическом представлении символа
010100111	011	110001	
010100111	100	111110	
010111111	101	101110	
010100111	110	110001	
010100111	111	111111	

А — коды символов
Б — номера строк растра в пределах знакомест

ски в течение развертки каждого телевизионного кадра выдавать последовательно все коды экранной области на адресные входы ПЗУ знакогенератора. Эту и ряд других функций и выполняет БИС контроллера ЭЛТ КР580ВГ75.

Включение контроллера ЭЛТ в систему

БИС КР580ВГ75 содержит в своем составе два 8-разрядных буферных регистра, в каждом из которых может храниться по 80 кодов символов для последующего их вывода на экран. В процессе работы, в то время как символы, хранимые в одном из буферных регистров, последовательно отображаются на экране дисплея, в другой буферный регистр переписываются из экранной области ОЗУ следующие символы. По окончании отображения символов из одного буферного регистра начинается процесс отображения из другого, т. е. регистры функционально меняются местами. Символы из экранной области ОЗУ в буферные регистры контроллера ЭЛТ пересылаются методом прямого доступа к памяти (ПДП). Поэтому работа контроллера ЭЛТ, как правило, осуществляется совместно с контроллером ПДП КР580ВТ57.

Назначение выводов микросхемы показано в табл. 2, а схема включения контроллера ЭЛТ в микропроцессорную систему представлена на рис. 2. Схема работает следующим образом. Контроллер ЭЛТ формирует сигнал «Запрос ПДП» на выводе DRQ в течение обратного хода кадровой развертки, раньше окончания кадрового гасящего импульса на период, равный длительности отображения одного ряда знакомест. В соответствии с этим запросом контроллер ПДП подготавливает систему к работе в режиме ПДП путем подачи сигнала на вход HOLD микропроцессора, который в ответ на него переводит свои шины в высокоимпедансное состояние, о чем и уведомляет контроллер ПДП сигналом HLDA. После этого контроллер ПДП устанавливает на выводе $\overline{DACK}$ контроллера ЭЛТ сигнал «Подтверждение ПДП». При наличии этого сигнала и при появлении сигнала «Запись ввода-вывода» от контроллера ПДП на выводе $\overline{WR}$ контроллера ЭЛТ байт с шины данных переписывается в буферный регистр. Сигнал «Запись ввода-вывода» вырабатывается контроллером ПДП для каждого очередного байта. Предварительно контроллер ПДП выдает на шину адресов адрес очередной ячейки экранной области ОЗУ и на шину управления сигнал «Чтение памяти», инициирующий вывод данных из ОЗУ на шину данных.

Последовательность символов, переписываемую в буферный регистр контроллера дисплея за время действия сигнала «Запрос ПДП», называют пакетом. Длительность этого сигнала зависит от числа символов в пакете, которое задается программно при начальной инициализации контроллера ЭЛТ и может быть равна 1, 2, 4 или 8. Пакеты следуют друг за другом с интервалом времени, который может быть запрограммирован в диапазоне от 0 до 55 T_{CCLK}. По истечении каждого такого интервала контроллер ЭЛТ вновь выдает сигнал «Запрос ПДП». Сигналы «Запрос ПДП» будут следовать, пока буферный регистр контроллера дисплея не заполнится (заполнение буфера может произойти и до окончания передачи пакета). Число символов в пакете и интервалы времени между ними задаются из системных соображений. Например, если процесс загрузки кодов символов в контроллер ЭЛТ одновременно используется в системе для регенерации динамического ОЗУ, то временные

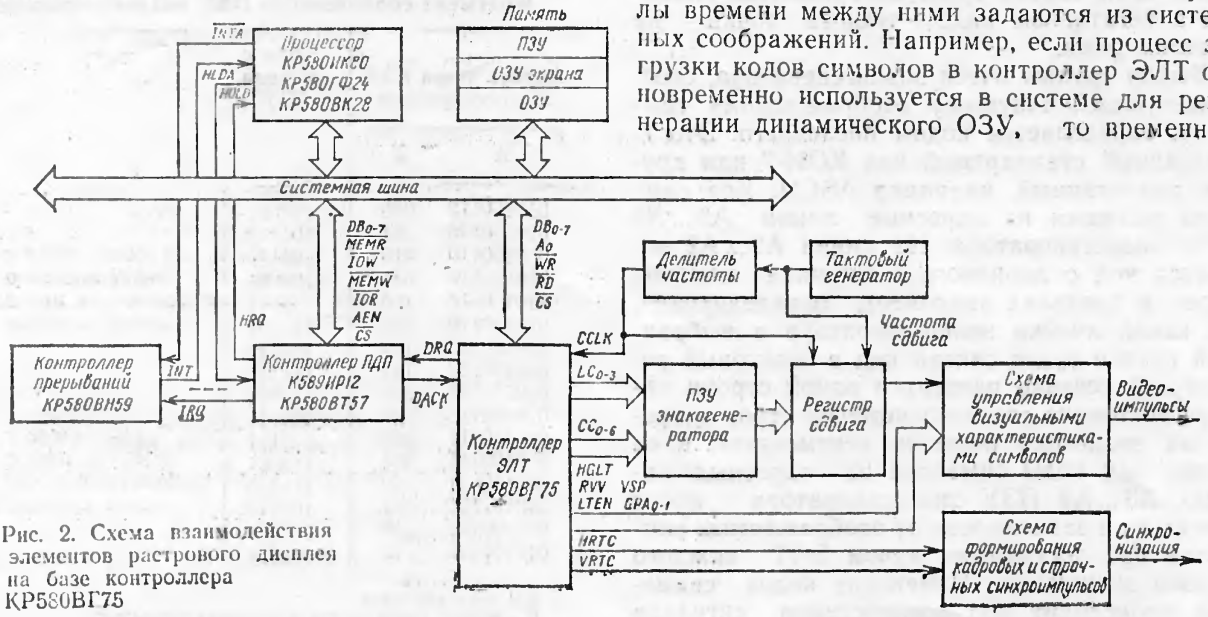


Рис. 2. Схема взаимодействия элементов растрового дисплея на базе контроллера КР580ВГ75

Таблица 2

Назначение выводов контроллера ЭЛТ КР580ВГ75

Номер вывода	Обозначение вывода		Наименование сигнала	Функциональное назначение сигнала
40	V_{cc}	Вход	Напряжение питания + 5 В	Примечание. Максимальный потребляемый ток 160 мА.
20	GND		Общий	
			Сигналы подключения к системным шинам	
12 ... 19	DB_0 ... DB_7	Двунаправленная шина данных	Линии шины данных	Прием команд, параметров, кодов и признаков в режиме ПДП. Выдача содержимого внутренних регистров контроллера. Переходит в высокоимпедансное состояние, если $\overline{RD} \wedge \overline{WR} = 1$ или $\overline{CS} = 1$
22	$\overline{CS}$	Вход	Выбор микросхемы	Разрешает считывание или запись байта
21	A_0	Вход	Адрес порта	Если $A_0 = 0$, то разрешен обмен с регистрами параметров; если $A_0 = 1$, то разрешено считывание из регистра состояния или запись в регистр команд
10	$\overline{WR}$	Вход	Запись	В программном режиме запись команд и параметров В режиме ПДП запись кодов символов и признаков При передаче пакета на этот вход поступают сигналы записи от контроллера ПДП. Число сигналов определяется длительностью действия сигнала «Запрос ПДП» и равно запрограммированному числу байтов в пакете
9	$\overline{RD}$	Вход	Считывание	Считывание параметров или слова состояния
			Сигналы управления ПДП	
31	IRQ	Выход	Запрос прерывания	Запрос на прерывание процессора. Если программно не запрещен, то возникает в начале отображения последнего в кадре ряда знакомест. Сбрасывается после считывания регистра состояния
5	DRQ	Выход	Запрос ПДП	Активизирует контроллер ПДП на время передачи пакета. Возникает во время обратного хода кадровой развертки и затем перед началом передачи каждого пакета. Сбрасывается по окончании передачи пакета или по заполнении буфера символов
6	$\overline{DACK}$	Вход	Подтверждение ПДП	Поступает от контроллера ПДП. Активен в течение действия сигнала «Запрос ПДП». Разрешает запись пакета в буфер контроллера
			Сигналы управления ПЗУ знакогенератора	
23 ... 29	CC_0 ... CC_6	Выходы	Код отображаемого символа	Подаются на адресные линии старших разрядов ПЗУ знакогенератора для выбора группы ячеек памяти, где хранится информация о графическом представлении символов на экране
1 ... 4	LC_3 ... LC_0	Выходы	Счетчик строк растра в пределах знакоместа	Подаются на адресные линии младших разрядов ПЗУ знакогенератора для выбора ячейки памяти внутри группы
			Сигналы управления разверткой и видеосигналом	
38, 39	LA_1, LA_0	Выходы	Признаки псевдографических символов	Управляют внешними схемами при отображении графических изображений в виде комбинаций вертикальных и горизонтальных линий (псевдографика)
7	HRTC	Выход	Обратный ход по строке	Используется для генерации строчных синхросигналов
8	VRTC	Выход	Обратный ход по кадру	Используется для генерации кадровых синхросигналов

Окончание табл. 2

Номер вывода	Обозначение вывода		Наименование сигнала	Функциональное назначение сигнала
35	VSP	Выход	Гашение луча	Сигнал активен: 1) если HRTC=1 или VRTC=1; 2) при затемнении верхней и нижней строк растра в пределах знакоместа, если запрограммировано подчеркивание строки с номером больше 8; 3) при появлении кода «Конец строки символов» или «Конец экрана»; 4) при недогрузке ПДП; 5) при отображении мигающих курсора или символов с частотой, равной соответственно 1/16 и 1/32 частоты кадровой развертки
37	LTEN	Выход	Разрешение видеосигнала	Активен в момент отображения курсора в виде подчеркивания, элементов псевдографики и символов с соответствующими признаками
36	RVV	Выход	Инвертирование видеосигнала	Активен при отображении алфавитно-цифровых и псевдографических символов с соответствующими признаками
33, 34	GPA_0, GPA_1	Выходы	Универсальные признаки	Активны при появлении соответствующих признаков. Служат для реализации каких-либо дополнительных функций, например отображения многоцветных символов на экране ЭЛТ цветного изображения
32	HLGT	Выход	Подсветка	Используется для увеличения яркости свечения символов при соответствующем признаке
			Прочие сигналы	
30	CCLK	Вход	Символьная синхронизация	Период следования сигналов определяется временем прохождения луча ЭЛТ в пределах знакоместа. Минимальный период следования 320 нс
11	LPEN	Вход	Световое перо	Активизирует запоминание в регистрах светового пера контроллера номера знакоместа в ряду и номера ряда. Используется для определения положения светового пера на экране

параметры ПДП выбираются исходя из условий регенерации памяти.

Контроллер ПДП при работе с контроллером ЭЛТ обычно программируется на работу в режиме с автозагрузкой. При этом используется канал 2, параметры которого каждый раз автоматически восстанавливаются после окончания процесса пересылки всего содержимого экранной области ОЗУ. Однако бывают особые случаи, когда по окончании отображения кадра требуется перезагрузка контроллера ПДП новыми параметрами, например при необходимости быстрого сдвига всего текста на экране на несколько строк вверх или вниз. В этом случае контроллер ЭЛТ может быть запрограммирован на формирование в конце каждого кадра сигнала «Запрос прерывания». По возникновении этого сигнала программа обслуживания прерывания может изменить параметры в контроллере ПДП.

Необходимо отметить, что использование режима ПДП для обслуживания контроллера ЭЛТ снижает общую производительность микропроцессорной системы. Оценку доли времени, используемой для обслуживания контроллера, нужно проводить исходя из числа знакомест на экране ЭЛТ, частоты смены кадров и времени, затрачиваемого на передачу одного байта в режиме ПДП (при использовании контроллера ПДП КР580ВТ57 передача первого байта в пакете осуществляется за пять-шесть машинных тактов, а последующих — за четыре).

Как уже указывалось, в буферные регистры контроллера ЭЛТ заносятся 8-разрядные коды. Однако контроллер рассчитан только на отображение символов, кодируемых 7-разрядными кодами, поэтому восьмой разряд кодов отображаемых символов должен быть равен нулю. Наличие единицы в восьмом разряде свидетельствует, что данный код является либо специальным кодом, либо визуальным признаком алфавитно-цифровых символов, либо признаком псевдографических символов.

Специальные коды (рис. 3) позволяют более экономно использовать временные ре-

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	1	0	0	S	S

Рис. 3. Специальные коды:
00 — Конец строки символов; 01 — Конец строки символов — Стоп ПДП; 10 — Конец экрана; 11 — Конец экрана — Стоп ПДП

сурсы и память микропроцессора, а также в некоторых случаях упростить программирование. Так, коды «Конец строки символов» и «Конец экрана» позволяют программисту не заботиться о стирании остатков прежнего текста на конце строки и экрана. Эти коды активизируют сигнал на выходе VSP контроллера. С помощью внешних схем этот сигнал используется для затемнения изображения, после того как в тексте встретятся специальные символы соответственно до конца строки или конца экрана.

Специальные коды «Конец строки символов» — «Останов ПДП» и «Конец экрана» — «Останов ПДП» приостанавливают режим ПДП, чтобы не считывать из экранного ОЗУ коды символов, которые не будут затем отображены. Если эти коды не являются последними в строке символов или в пакете, то дополнительно в буфер контроллера ЭЛТ считывается следующий за специальным кодом символ.

Часто на экране необходимо визуально выделять один или несколько символов подчеркиванием или путем отображения их с повышенной яркостью, в негативном виде или мигающими. Для этого в схеме дисплея необходимо иметь дополнительные элементы, которые реализуют эти функции и управляются с помощью сигналов от контроллера ЭЛТ.

Перед символом или группой символов, которые необходимо визуально выделить на экране, в экранной области ОЗУ записывают соответствующий код визуального признака. Действие этого признака распространяется на все символы, следующие вслед за ним до конца экрана или до появления в экранной области ОЗУ нового кода признака. Возможно использование следующих признаков: мигания, подсветки, отображения символа в негативном виде, подчеркивания. Эти признаки активизируют выходные сигналы соответственно на выходах VSP, HLGT, RVV и LTEN контроллера. Имеется также возможность с помощью еще двух универсальных признаков выдать сигналы на выходе GPA0 и GPA1. Назначение этих сигналов произвольное и может дополнительно определяться разработчиками дисплея (рис. 4).

В зависимости от того, как запрограммирован контроллер ЭЛТ, признаки, встречающиеся в экранной области ОЗУ, могут быть не отображены вообще или отображены в виде пробела. Выбор одного из этих режимов определяет, будет ли всегда соблюдаться однозначное соответствие адресов экранной области ОЗУ положению знакомест на экране ЭЛТ (когда признаки отображаются в виде пробелов) или же такое соответствие будет динамически изменяться в соответствии с числом и расположением признаков в экранной области ОЗУ (в случае, когда признаки не отображаются). Во втором случае, если из экранной области ОЗУ в контроллер будет передан код признака, то он будет записан в один из его 80-символьных буферных регист-

D7	D6	D5	D4	D3	D2	D1	D0
1	1	C	C	C	C	B	H

Рис. 4. Признаки алфавитно-цифровых символов:
H=1 — подсветка (сигнал на выходе HLGT); B=1 — мигание символа (периодически появляется сигнал на выходе VSP); G_0=1 — универсальный (сигнал на выходе GPA_0); G_1=1 — универсальный (сигнал на выходе GPA); R=1 — инвертирование видеосигнала (сигнал на выходе RVV); U=1 — подчеркивание (сигнал на выходе LTEN)

а)

D7	D6	D5	D4	D3	D2	D1	D0
1	C	U	R	G1	G0	B	H

б)

Биты CCCC	Сигналы на выходах LA_1	LA_0	VSP	LTEN	Отображаемый символ
0000	0	0	1	0	
	1	0	0	0	
	0	1	0	0	
0001	0	0	1	0	
	1	1	0	0	
	0	1	0	0	
0010	0	1	0	0	
	1	0	0	0	
	0	0	1	0	
0011	0	1	0	0	
	1	1	0	0	
	0	0	1	0	
0100	0	0	1	0	
	0	0	0	1	
	0	1	0	0	
0101	0	1	0	0	
	1	1	0	0	
	0	1	0	0	
0110	0	1	0	0	
	1	0	0	0	
	0	1	0	0	
0111	0	1	0	0	
	0	0	0	1	
	0	0	1	0	
1000	0	0	1	0	
	0	0	0	1	
	0	0	1	0	
1001	0	1	0	0	
	0	1	0	0	
	0	1	0	0	
1010	0	1	0	0	
	0	0	0	1	
	0	1	0	0	

Рис. 5. Коды псевдографических символов:
CCCC — код псевдографического символа; H=1 — подсветка (сигнал на выходе HLGT); B=1 — мигание символа (периодически появляется сигнал на выходе VSP) (*а*), комбинации кодов и отображаемый символ (*б*). Остальные комбинации для псевдографических символов не используются

ров. Но при этом следующий за признаком алфавитно-цифровой символ будет записан не в этот буферный регистр, а в специальный 7-разрядный стековый регистр. Таких регистров в контроллере два — по числу буферных регистров. В каждый из стековых регистров может быть записано не более 16 кодов символов. Стек организован по принципу «первый вошел — первый вышел», поэтому при выводе символов на экран из буферного регистра в случае появления среди них кода признака последний не отображается, а отображается вместо него очередной код символа из стекового регистра.

Такая организация контроллера накладывает следующие *ограничения на использование кодов признаков.* Во-первых, их не должно быть более 16 в одном ряде знакомест. Во-вторых, нельзя допускать, чтобы два признака непосредственно следовали друг за другом — в этом случае 8-разрядный код второго признака будет отправлен в стековый регистр, где он «потеряет» старший бит и в дальнейшем будет ошибочно отображаться как символ.

Контроллер ЭЛТ позволяет под управлением кодов псевдографических символов (рис. 5, *а, б*) *отображать на экране комбинации вертикальных и горизонтальных линий.* В правой части рис. 5, *б* показаны элементы, из которых состоят такие изображения. Каждый элемент — псевдографический символ, так же как и алфавитно-цифровой символ, располагается в пределах знакоместа.

Для отображения псевдографических символов вместо ПЗУ знакогенератора используется специальная дополнительная схема (рис. 6). При появлении хотя бы на одном из выходов LA_1, LA_0 контроллера «Лог. 1», выходы ПЗУ знакогенератора отключаются от входов сдвигового регистра D7. В зависимости от комбинации на выходах LA_1, LA_0, VSP и LTEN отображается один из псевдографических символов (см. рис. 5, 6). Каждое знакоместо при его отображении можно рассматривать как состоящее из трех частей — средней части в виде линии подчеркивания и частей, расположенных над и под линией подчеркивания. Для отображения каждой из этих частей знакоместа на выходы LA_1, LA_0, VSP и LTEN выдается соответствующий код, поэтому одному коду псевдографических символов соответствуют три кодовых комбинации. У выходов элементов D4.1...D4.3 на рис. 6 условно показаны те части символов, которые формируются на экране при их активизации. Изображение полной горизонтальной линии в середине знакоместа происходит за счет появления сигнала на выходе LTEN.

Каждый псевдографический символ может

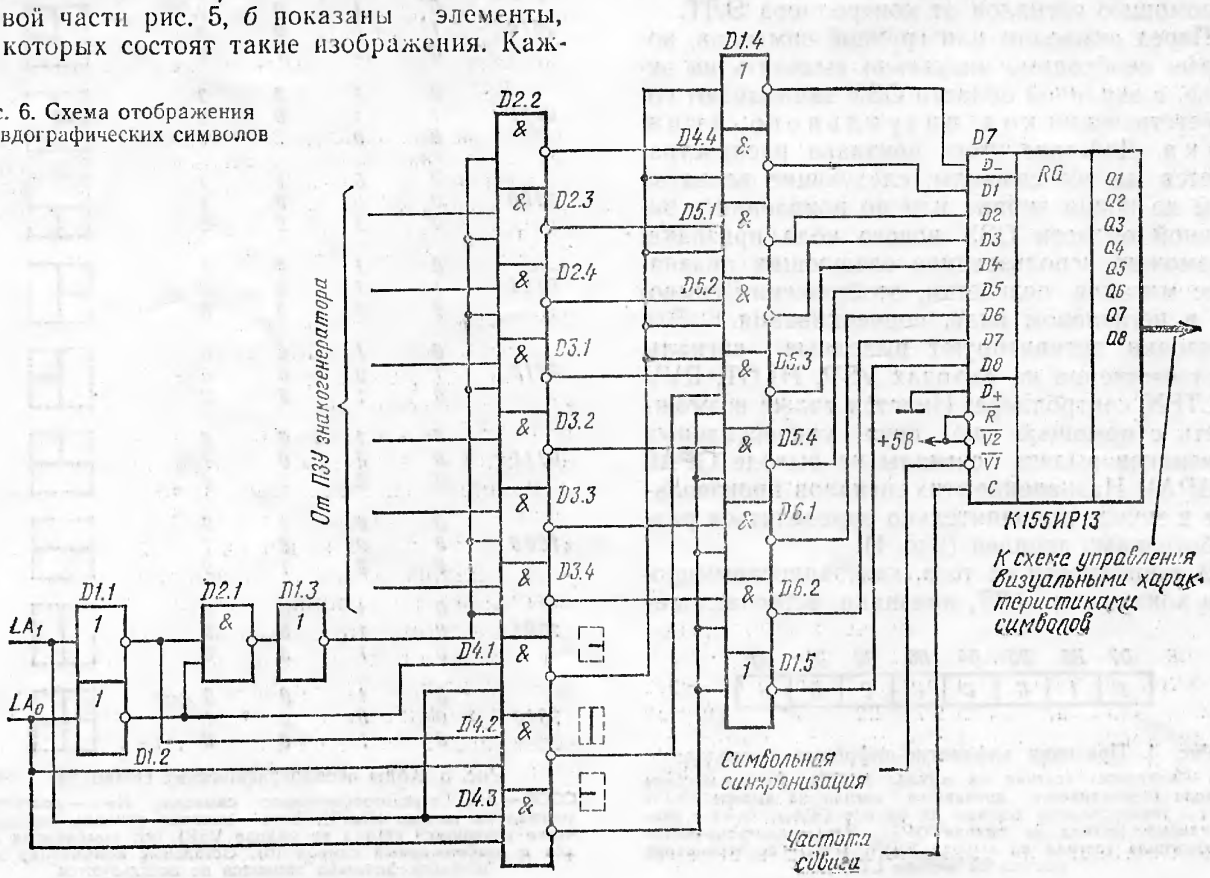


Рис. 6. Схема отображения псевдографических символов

быть визуально выделен на экране повышенной яркостью или миганием с помощью задания значения двух младших битов в коде псевдографического символа (см. рис. 5). На визуальное выделение псевдографических символов влияют также визуальные признаки алфавитно-цифровых символов — признак отображения символа в негативном виде и универсальные признаки.

Курсор и световое перо

При работе с дисплеем необходимо помечать то знакоместо, куда будет выведен очередной символ. Это осуществляется с помощью специального знака — курсора. БИС контроллера ЭЛТ позволяет запрограммировать вывод курсора на экран в одном из четырех видов: мигающей линии подчеркивания; мигающего символа (или знакоместа) в негативном виде; немигающей линии подчеркивания; немигающего алфавитно-цифрового символа (или знакоместа) в негативном виде. Формирование курсора осуществляется с помощью дополнительных схемных элементов, управляемых сигналами VSP, RVV и LTEN.

Контроллер ЭЛТ позволяет подключать к дисплею световое перо для непосредственного указания на экране какого-либо знакоместа. Световое перо обычно представляет из себя небольшой фотодатчик, который прижимается к поверхности экрана. При этом микропереключатель, вмонтированный в корпус датчика, замыкает цепь, и фотодатчик оказывается подключенным к входу LPEN контроллера. Когда луч ЭЛТ при развертке попадает на фотодатчик, то последний вырабатывает сигнал, по которому в регистрах светового пера контроллера запоминаются координаты положения засветившего датчика символа. Состояние этих регистров может быть считано по команде «Считывание положения светового пера». Из-за задержек прохождения сигнала от светового пера в контроллере и внешних схемах координаты фотодатчика считываются с ошибкой (более, чем на три знакоместа). Это должно быть скорректировано программным обеспечением.

Программирование контроллера

Программирование осуществляется путем записи команд и параметров во внутренние регистры команд и параметров. Всего имеется восемь команд, некоторые из них требуют ввода дополнительных параметров (табл. 3). Вслед за командой «Считывание положения светового пера» *требуется считывать* (не записывать!) два параметра. Параметры должны следовать непосредственно друг за другом. Ниже приведен набор программируемых параметров контроллера.

Таблица 3

Команды управления контроллером

Команда, параметр	A0	D7	D6	D5	D4	D3	D2	D1	D0
Формат экрана	1	0	0	0	0	0	0	0	0
Параметр 1	0	S	H	H	H	H	H	H	H
Параметр 2	0	V	V	R	R	R	R	R	R
Параметр 3	0	U	U	U	U	L	L	L	L
Параметр 4	0	M	F	C	C	Z	Z	Z	Z
Разрешение отображения	1	0	0	1	S	S	S	B	B
Запрет отображения	1	0	1	0	0	0	0	0	0
Управление курсором	1	1	0	0	0	0	0	0	0
Параметр 1	0	Номер знакоместа в ряду							
Параметр 2	0	Номер ряда знакомест							
Разрешение прерывания	1	1	0	1	0	0	0	0	0
Запрещение прерывания	1	1	1	0	0	0	0	0	0
Предварительная установка счетчиков	1	1	1	1	0	0	0	0	0
Считывание положения светового пера	1	0	1	1	0	0	0	0	0
Параметр 1	0	Номер знакоместа в ряду							
Параметр 2	0	Номер ряда знакомест							

Параметры всех команд загружаются в контроллер непосредственно вслед за загрузкой соответствующих команд. Исключение составляет команда «Считывание положения светового пера», параметры 1 и 2 которой считываются из контроллера непосредственно вслед за загрузкой команды

Поясним режим счета строк растра в знакоместе. *Режим без смещения* применяется для ПЗУ знакогенераторов, в которых начальные ячейки в группах с информацией о графическом представлении символов используются для затемнения строки. Если же в этих ячейках записана информация об отображаемом символе, то используется *режим счета со смещением* на единицу (см. рис. 1). Рассмотрим действия, выполняемые командами.

Команда «Формат экрана» прекращает процесс ПДП (прохождение сигнала «Запрос прерывания»), сигнал на выходе VSP используется для затемнения экрана. Сигналы на выходах HRTC и VRTC находятся в активном состоянии.

Команда «Разрешение отображения» разрешает прохождение сигналов «За-

Отображение рядов знакомест — бит S

0 ..... Отображаются все ряды

1 ..... Отображаются только четные ряды

Число знакомест в ряду — биты НННННН

0000000 1
0000001 2
0000010 3
... ...
1001111 80

Остальные кодовые комбинации не используются

Число рядов знакомест в кадре — биты RRRRRR

000000 1
000001 2
000010 3
... ...
111111 64

Число строк растра в знакоместе — биты LLLL

0000 1
0001 2
0010 3
... ...
1111 16

Длительность обратного хода кадровой развертки — биты VV

00 ..... 1 T
01 ..... 2 T
10 ..... 3 T
11 ..... 4 T

T — время отображения одного ряда знакомест.

Номер линии подчеркивания — биты UUUU

0000 1
0001 2
0010 3
... ...
1111 16

Режим счета строк растра в знакоместе — бит M

0 ..... Режим без смещения

1 ..... Режим со смещением на 1

Режим отображения признаков — бит F

0 ..... В виде пробела

1 .... Не отображаются

ПДП — биты BB

Число байт в пакете ПДП — биты BB

00 1
01 2
10 4
11 8

Формат курсора — биты CC

00 ..... Мигающий символ

01 ..... Мигающая линия подчеркивания

10 ..... Немигающий символ в негативном виде

11 ..... Немигающая линия подчеркивания

Длительность обратного хода строчной развертки — биты ZZZZ

0000 ..... 2Tсclk
0001 ..... 4Tсclk
0010 ..... 6Tсclk
... ...
1111 ..... 32Tсclk

Время между запросами ПДП — биты SSS

000 ..... 0Tсclk
001 ..... 7Tсclk
010 ..... 15Tсclk
011 ..... 23Tсclk
100 ..... 31Tсclk
101 ..... 39Tсclk
110 ..... 47Tсclk
111 ..... 55Tсclk

прос прерывания» и «Запрос ПДП». При этом устанавливаются соответствующие биты в слове состояния и разрешается появление изображения на экране.

Команда «Конец отображения» запрещает прохождение видеосигнала. Прохождение сигналов «Запрос прерываний» и «Запрос ПДП» продолжает оставаться разрешенным, сигналы обратного хода луча по строкам и кадрам продолжают проходить.

По команде «Считывание сигнала светового пера» из внутренних регистров светового пера контроллера считываются ранее записанные координаты пера на экране — номер знакоместа и номер ряда.

Команда «Загрузка положения

```
;---------- ИНИЦИАЛИЗАЦИЯ КОНТРОЛЛЕРОВ ----------
PORT75  EQU     10H     ;АДРЕС ПОРТА ЗАГРУЗКИ ПАРАМЕТРОВ КОНТРОЛЛЕРА ЭЛТ
PORT67  EQU     20H     ;АДРЕС ПОРТА ЗАГРУЗКИ ПАРАМЕТРОВ КАНАЛА 0 ПДП

SR      EQU     0       ;ОТОБРАЖАЮТСЯ ВСЕ РЯДЫ
VRTC    EQU     1       ;ДЛИТЕЛЬНОСТЬ ОБРАТНОГО ХОДА КАДРОВОЙ РАЗВЕРТКИ
WIDTH   EQU     78      ;ЧИСЛО СИМВОЛОВ В РЯДУ
HEIGHT  EQU     30      ;ЧИСЛО РЯДОВ НА ЭКРАНЕ
UL      EQU     10      ;НОМЕР ЛИНИИ ПОДЧЕРКИВАНИЯ
LN      EQU     10      ;ЧИСЛО ЛИНИЙ В СИМВОЛЕ
LM      EQU     1       ;РЕЖИМ СЧЕТА СТРОК РАСТРА СО СМЕЩЕНИЕМ
AM      EQU     0       ;ОТОБРАЖАЕМЫЕ ВИЗУАЛЬНЫЕ ПРИЗНАКИ
CF      EQU     1       ;ФОРМАТ КУРСОРА - МИГАЮЩАЯ ЛИНИЯ ПОДЧЕРКИВАНИЯ
HRTC    EQU     8       ;ДЛИТЕЛЬНОСТЬ ОБРАТНОГО ХОДА СТРОЧНОЙ РАЗВЕРТКИ
DRT     EQU     000000011B ;ВРЕМЯ МЕЖДУ ЗАПРОСАМИ ПДП = 7
BSZ     EQU     000000111B ;ЧИСЛО БАЙТ В ПАКЕТЕ ПДП = 8
PDPMODE EQU     100000000B ;РЕЖИМ ЧТЕНИЯ ПДП

;---------- КОМАНДА ФОРМАТ ЭКРАНА ----------
        MVI     A,0
        OUT     PORT75+1
        MVI     A,(SR SHL 7) OR WIDTH-1
        OUT     PORT75
        MVI     A,((VRTC-1) SHL 6) OR (HEIGHT-1)
        OUT     PORT75
        MVI     A,(UL-1)*16+LN-1
        OUT     PORT75
        MVI     A,(LM SHL 7) OR (AM SHL 6) OR (CF SHL 5) OR (HRTC/2-1)
        OUT     PORT75
;---------- КОМАНДА РАЗРЕШЕНИЕ ОТОБРАЖЕНИЯ ----------
        MVI     A,20H+(DRT SHL 2)+03
        OUT     PORT75+1
;---------- НАСТРОЙКА КОНТРОЛЛЕРА ПДП ----------
        MVI     A,80H           ;УСТАНОВИТЬ БИТ АВТОЗАГРУЗКИ
        OUT     PORT67+8
        MVI     A,BUF AND 0FFH  ;ЗАГРУЗИТЬ МЛАДШИЙ БАЙТ АДРЕСА
        OUT     PORT67-4        ;ОЗУ ЭКРАНА
        MVI     A,BUF SHL 8     ;ЗАГРУЗИТЬ СТАРШИЙ БАЙТ АДРЕСА
        OUT     PORT67+4        ;ОЗУ ЭКРАНА
BUFSZ   SET     WIDTH*HEIGHT    ;ВЫЧИСЛЕНИЕ РАЗМЕРА ОТОБРАЖАЕМОЙ ОБЛАСТИ
        MVI     A,(BUFSZ-1) AND 0FFH ;ЗАГРУЗКА МЛАДШЕГО БАЙТА СЧЕТЧИКА
        OUT     PORT67-8        ;ПЕРЕДАВАЕМОЙ ОБЛАСТИ
        MVI     A,PDPMODE OR ((BUFSZ-1) SHL 8) ;УСТАНОВКА РЕЖИМА ЧТЕНИЯ
        OUT     PORT67-5        ;ПДП И ЗАГРУЗКА СТАРШЕГО БАЙТА СЧЕТЧИКА
;---------- КОМАНДА ПРЕДУСТАНОВКИ СЧЕТЧИКОВ ----------
        MVI     A,08H
        OUT     PORT75+2
;---------- КОМАНДА ЗАПУСК ПДП ----------
        MVI     A,84H           ;УСТАНОВИТЬ БИТ АВТОЗАГРУЗКИ
        OUT     PORT67+8        ;И РАЗРЕШИТЬ КАНАЛ 2
```

ПРОГРАММА ИНИЦИАЛИЗАЦИИ КОНТРОЛЛЕРОВ

указателя курсора» позволяет загрузить два байта параметров — положение курсора. На регистр состояния команда не влияет.

Команды «Разрешение прерывания» и «Запрещение прерывания» соответственно устанавливают режим, разрешающий и запрещающий возникновение запросов прерывания и управляют соответствующим битом слова состояния.

Команда «Предустановка счетчиков» позволяет установить счетчики знакомест в состояние, соответствующее отображению символа в верхнем левом углу дисплея. Для этой операции требуются два символьных синхроимпульса. Счетчики будут оставаться в таком состоянии до тех пор, пока не будет подана любая следующая команда. Команда необходима, когда в системе работают несколько контроллеров ЭЛТ, работу которых необходимо синхронизировать.

Для программного анализа состояния, в котором находится контроллер ЭЛТ, в процессе работы можно считывать из регистра состояния слово состояния контроллера. Назначение отдельных битов регистра состояния контроллера следующее:

D6 — Разрешение прерывания. Устанавливается с помощью команд «Разрешение прерывания» и «Разрешение отображения», а также при включении питания. Сбрасывается командами «Запрещение прерывания» и «Формат экрана».

D5 — Запрос на прерывание. Устанавливается в начале отображения последнего ряда

знакомест кадра, если было разрешено прерывание. Сбрасывается при считывании слова состояния.

D4 — Сигнал от светового пера. Сбрасывается при считывании слова состояния.

D3 — Неверная команда. Устанавливается в случае несоответствия команде числа загруженных параметров или недопустимого кода. Сбрасывается считыванием слова состояния.

D2 — Разрешение отображения. Устанавливается по команде «Разрешение отображения» и сбрасывается по командам «Запрет отображения» и «Формат экрана».

D1 — Недогрузка ПДП. Устанавливается при недогрузке данных в ходе отображения информации на экране. При появлении бита режим ПДП прекращается и экран затемняется до начала отображения нового кадра изображения. Сбрасывается при считывании слова состояния.

D0 — Переполнение стека символов. Устанавливается, если число признаков в строке символов превысило 16. Сбрасывается при считывании слова состояния.

Бит D7 в регистре всегда находится в состоянии «Лог. 0».

Пример реализации дисплея

При конструировании дисплея на базе контроллера ЭЛТ необходимо учитывать, что его параметры: *формат экрана* (число рядов знакомест и число знакомест в ряду), *формат знакомест* (размер матрицы засвечиваемых точек), *параметры развертки* (число кадров в секунду, число строк в кадре, длительности обратного хода кадровой и строчных разверток), *параметры синхроимпульсов* не могут выбираться произвольно.

На рис. 7 приведена схема формирования видеосигнала для простейшего дисплея, в качестве которого может быть использован телевизор. Программа инициализации контроллера дисплея приведена на с. 68. Предполагается, что на экране телевизора алфавитно-цифровые символы будут отображаться в виде 25 строк по 64 символа. Так как по краям экрана четкость изображения в телевизорах существенно меньше, чем в центре, то для отображения символов их использовать нельзя.

Для того чтобы иметь возможность непосредственно использовать сигналы HRTC и VRTC в качестве строчных и кадровых синхро-

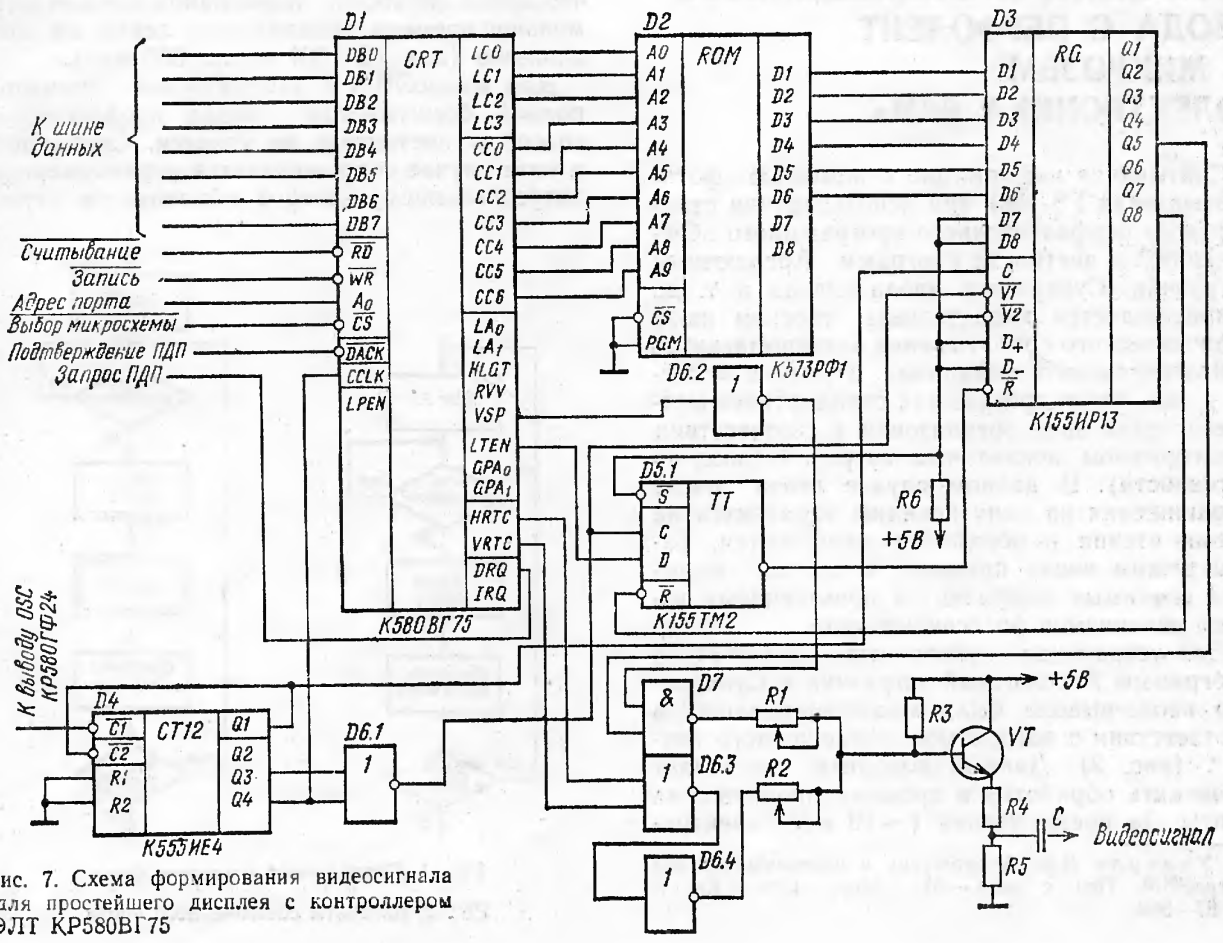


Рис. 7. Схема формирования видеосигнала для простейшего дисплея с контроллером ЭЛТ КР580ВГ75

импульсов, затемнение неиспользуемых краев экрана осуществляется программно путем записи кодов пробелов в ячейки экранной области ОЗУ, соответствующие краям экрана. При этом контроллер программируется на отображение формата 78×30 знакомест. Для простоты никакие визуальные признаки в данной схеме не используются. Курсор отображается в виде мигающей линии подчеркивания.

В высококачественных дисплеях необходимо учитывать задержки появления кода на выходе ПЗУ знакогенератора относительно заднего фронта сигнала на входе CCLK контроллера. Эта величина складывается из задержки появления кода символов на выходе контроллера ЭЛТ относительно отрицательного фронта символьного синхроимпульса и задержки появления кода на выходе ПЗУ знакогенератора после появления сигналов на его адресных шинах. А так как задержка прохождения остальных сигналов управления с выходов контроллера обычно оказывается меньше, то необходимо принимать меры по синхронизации всех сигналов с выходов контроллера.

Это можно осуществить, например, путем введения буферного регистра между контроллером ЭЛТ и ПЗУ знакогенератора. В этот буферный регистр коды символов записываются после положительного перепада импульса «Символьная синхронизация», а код из ПЗУ знакогенератора загружается в сдвиговый регистр при его следующем положительном перепаде. Таким образом, код с выхода сдвигового регистра будет задержан на 1,5 периода следования импульсов «Символьная синхронизация». В связи с этим необходима буферизация и соответствующая синхронизация других управляющих сигналов контроллера ЭЛТ.

Статья поступила 12 апреля 1985 г.

УДК 681.322.074

А. М. Буленков, Н. В. Буробин, Н. И. Титчев

ОРГАНИЗАЦИЯ СОВМЕЩЕННОГО ВВОДА С ПЕРФОЛЕНТ В МИКРОЭВМ «ЭЛЕКТРОНИКА 60М»

Считывание информации с помощью фотосчитывателя FS-1501 при использовании стандартного перфоленточного программного обеспечения, в частности программ Абсолютный загрузчик, Супервизор ввода-вывода и т. д., сопровождается характерным треском из-за периодического срабатывания электромагнитов лентопротяжного механизма. Причина состоит в том, что в программах стандартного матобеспечения ввод организован в соответствии с алгоритмом, показанным на рис. 1 (ввод по готовности). В данном случае лента после продвижения на одну позицию тормозится на время чтения и обработки информации. Такой режим ввода приводит к частым порывам вводимых перфолент и повышенному износу механизмов фотосчитывателя.

Для устранения указанных недостатков программы Абсолютный загрузчик и Супервизор ввода-вывода были модифицированы в соответствии с алгоритмом совмещенного ввода* (рис. 2). Данный алгоритм позволяет проводить обработку в процессе продвижения ленты. За время чтения (~10 мкс) электромагнит тормоза сработать не успевает, следовательно, перфолента не подвергается механическим нагрузкам и ввод становится бесшумным. Необходимо, однако, иметь в виду, что время обработки информации должно быть меньше времени продвижения ленты на одну позицию (для FS-1501 около 660 мкс).

Для микроЭВМ с динамической памятью полной бесшумности ввода предлагаемым способом достигнуть не удается. Считывание в этом случае сопровождается неравномерным потрескиванием, которое объясняется перио-

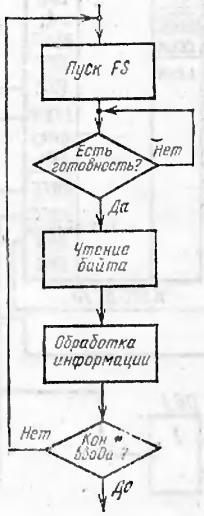


Рис. 1. Стандартный алгоритм ввода



Рис. 2. Алгоритм совмещенного ввода

* Уокерли Дж. Архитектура и программирование микроЭВМ: Пер. с англ.— М.: Мир, 1984.— Кн. 1, с. 387—389.

дическими прерываниями для регенерации памяти. Период регенерации, составляющий около 2 мс, не равен времени продвижения ленты на одну позицию. Прерывания могут возникать также во время операции чтения, а суммарного времени чтения и регенерации, составляющего 160... 170 мкс, вполне достаточно для срабатывания электромагнита тормоза.

Чтобы устранить это явление, следует синхронизировать начало регенерации памяти с началом продвижения перфоленты (рис. 3), подав сигнал «Старт» с выхода элемента D40.3 устройства управления В1 на вход генератора синхроимпульсов регенерации центрального процессора (вывод 1 микросхемы D57.1). Следует также иметь в виду, что при введении синхронизации допустимое время обработки информации сокращается на время регенерации, т. е. примерно до 500 мкс. Данное условие всегда выполняется при работе программы Абсолютный загрузчик.

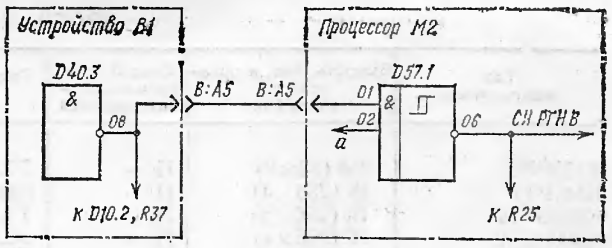


Рис. 3. Принципиальная схема организации совмещенного ввода для микроЭВМ с динамической памятью

Предлагаемая простая доработка позволила повысить скорость ввода с перфолент, продлить срок их службы более чем в 10 раз, устранить акустический шум, возникающий при вводе и снизить механический износ деталей фотосчитывателя.

Статья поступила 5 апреля 1985 г.

УЧИСЬ РАБОТАТЬ С ПЗУ

Статьи под рубрикой «Учись работать с ПЗУ» последовательно знакомят с физическими основами, методикой и способами записи информации в однократно программируемые и репрограммируемые ПЗУ (с электрическим стиранием и стиранием ультрафиолетовым излучением), а также со схемотехническими особенностями программаторов, позволяющих заносить (и считывать) информацию в ПЗУ различных типов.

Введение.

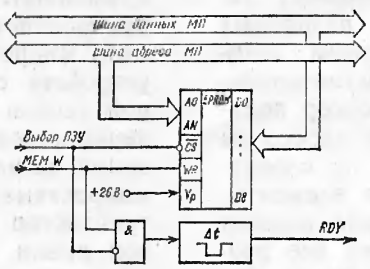
Постоянные запоминающие устройства (ПЗУ) — это не только устройства памяти программ микропроцессорных средств и микроЭВМ, но и формирователи заданных импульсных последовательностей, генераторы аналоговых сигналов необходимой формы, преобразователи кодов, дешифраторы и т. п. Они находят применение в различных областях радиоэлектроники.

Среди ПЗУ особенно выделяются электрически программируемые микросхемы, в которые разработчик может самостоятельно записать необходимую ему в конкретном проекте информацию. Несколько корпусов таких БИС ПЗУ способны заменить сложную и нерегулярную логическую схему из отдельных вентилей, что заметно сокращает время разработки и оставляет возможность модификации аппаратуры без изменения ее конструкции.

На практике наиболее часто применяются однократно программируемые постоянные запоминающие устройства (ППЗУ) с перемычками, разрушаемыми в процессе записи информации, и репрограммируемые ПЗУ (РПЗУ) с электрической записью и стиранием информации (ЭРПЗУ) и стиранием ультрафиолетовым излучением (УФРПЗУ).

Разнообразие типов ППЗУ (см. таблицу) и технологий программирования требует столь же разнообразных устройств для занесения информации (программаторов), не всегда доступных, если учесть опережающее развитие микроэлектроники. Это часто приводит к неоправданной ориентации на отдельные, нередко дорогие и дефицитные типы микросхем ПЗУ, ограничивает возможность замены ими нерегулярной логики, только из-за того, что имеется программатор, рассчитанный на определенный тип БИС.

Общая тенденция разработок новых БИС ПЗУ — упрощение процедуры занесения информации в микросхему благодаря размещению формирователей программирующих сигналов непосредственно на кристалле, если они рассеивают небольшую мощность. Например, информацию в БИС серии К573 (кроме РФ1 и РФ3) может заносить тот же микропроцессор, с которым будет работать микросхема. Для этого, кроме шин данных и адреса, к ней подводятся дополнительные сигналы (см. рисунок) и организуется обращение к области ПЗУ как к обычному ОЗУ в режиме записи, но цикл обращения удлиняется до 50 мс подачей сигнала подтверждения обмена на вход RDY микропроцессора.



Запись в ПЗУ непосредственно в составе МП-аппаратуры

Для других типов ПЗУ необходимы специальные устройства — программаторы. Простейшие программаторы обычно не содержат буферной памяти, и обеспечивают копирование информации из «эталонного» ПЗУ в один-два типа микросхем. Такие устройства наиболее удобны в условиях массового производства МП-аппаратуры. Для серьезной работы с ПЗУ в лаборатории необходим универсальный прибор, позволяющий заносить и считывать информацию как минимум во все типы БИС, перечисленные в таблице, и блоки на их основе. Его функции проще всего реализуются встроенными или внешними микропроцессорами или ЭВМ.

В статьях двух выпусков «Учебного центра» будут последовательно рассмотрены физические основы занесения информации в ПЗУ, методика записи и необходимая для этого аппаратура (программаторы), нетрадиционные применения ПЗУ в электронной аппаратуре. Хотя рассматриваемые программаторы предназначены для работы с различными ЭВМ, использование внутреннего канала обмена информацией МП КР580ИК80 позволяет подключать их к любой ЭВМ с соответствующим промежуточным интерфейсом.

Тип микросхемы	Емкость, бит, и организация ячеек×бит	Способ программирования/стирания	Технология, время доступа, нс	Напряжение питания, В	Выходы информации	Тип корпуса
К155РЕ3	256 (32×8)	П/—	ТТЛ, 50	+5	ОК	238.16—2
К541РТ1	1К (256×4)	П/—	ИИЛ, 70	+5	3с	402.16—21
КР556РТ4	1К (256×4)	П/—	ТТЛШ, 90	+5	ОК	238.16—2
К500РЕ149	1К (256×4)	П/—	ЭСЛ, 35	—5,2	—	288.16—2
КР556РТ5	4К (512×8)	П/—	ТТЛШ, 70	+5	3с	239.24—2
КМ556РТ6	16К (2К×8)	П/—	ТТЛШ, 80	+5	ОК	210Б.24—1
КМ556РТ7	16К (2К×8)	П/—	ТТЛШ, 80	+5	3с	210Б.24—1
КР1601РР1*	4К (1К×4)	Э/Э	МОП, 1700	+5, —12	3с	
К558РР1	1К (256×4)	Э/Э	МОП, 1000	+5, —12	3с	
К573РФ1*	8К (1К×8)	Э/УФ	ЛИЗМОП, 450	+12, +5, —5	3с	210Б.24—5
К573РФ2*	16К (2К×8)	Э/УФ	ЛИЗМОП, 350	+5	3с	210Б.24—5
К573РФ5	16К (2К×8)	Э/УФ	ЛИЗМОП, 450	+5	3с	210Б.24—5
К573РФ4*	64К (8К×8)	Э/УФ	ЛИЗМОП, 450	+5	3с	2121.28—2
К573РФ6	64К (8К×8)	Э/УФ	ЛИЗМОП, 450	+5	3с	2121.28—2

* Микросхемы выпускаются и с меньшей информационной емкостью. В этом случае в обозначение вводится цифра: 1, 2—допустимо использование только половины адресов; 3, 4— допустимо использование только половины информационных выходов. Например, К573РФ21—1Кх8, К573РФ24—2Кх4.

Обозначение: П — программирование прожиганием перемычек в ячейках БИС; Э — электрическая запись (стирание) информации; УФ — стирание информации ультрафиолетовым излучением.

УДК 681.326.3:681.513.2

О. А. Щербаков

ФИЗИЧЕСКИЕ ОСНОВЫ ЗАПИСИ ИНФОРМАЦИИ В ПЗУ

Среди разнообразных типов ПЗУ, программируемых пользователем, наибольшее распространение получили ПЗУ с ультрафиолетовым стиранием серии К573, ПЗУ с плавкими перемычками серий К556 и К541, ПЗУ с электрическим стиранием и записью информации серий К558, К1601, К1609.

Во всех перечисленных типах запоминающих устройств элементы, хранящие информацию, располагаются в виде ячеек двумерной матрицы. Каждая ячейка может хранить один бит информации, т. е. быть в состояниях «Лог. 0» и «Лог. 1». Физически на кристалле микросхемы ПЗУ ячейки располагаются на пересечении так называемых *словарных линий*, идущих от дешифратора, и *разрядных линий*, перпендикулярных словарным, которые подсоединяются по входам мультиплексора. На дешифратор и мультиплексор подаются разряды адреса. При подаче адреса на дешифратор возбуждается одна из словарных линий и все запоминающие элементы, расположенные на ней, параллельно выдают хранящуюся в них информацию на все разрядные линии. Выборка нужного числа бит для подачи на выход микросхемы ЗУ осуществляется мультиплексором. В зависимости от организации микросхемы, мультиплексор и дешифратор могут иметь различную разрядность. Например, микросхема емкостью 2К×8 может быть организована как матрица размером 128×128, что означает использование внутри микросхемы дешифратора «1 — из — 128» для возбуждения словарных линий и восьми мультиплексоров «16 — в — 1» для считывания разрядных линий.

С учетом топологических и технологических особенностей каждого типа микросхем может производиться деление матрицы запоминающих ячеек на блоки других размеров. Подобное построение запоминающих устройств является общим для всех типов. Отличия между ними заключаются в организации запоминающих ячеек, располагающихся на пересечении словарной и разрядных линий.

Микросхемы с плавкими перемычками. Выполненные по ТТЛ- или ТТЛШ-технологии, микросхемы данного типа применяются там, где необходимо высокое быстродействие. На их основе создается память микропрограмм для микропроцессорных устройств с разрядно-модульной архитектурой (серии К589, К1802, К1804), устройства перемножения и функционального преобразования сигналов. Запоминающим элементом в микросхемах данного типа является $n—p—n$ транзистор, подсоединенный базой к словарной линии, коллектором к $+u_п$, а эмиттером, через плавкую перемычку, к разрядной линии. В качестве плавкой перемычки используется поликристаллический кремний

или нихром, напыленные в процессе изготовления микросхемы.

Протекание тока программирования через нихромовую перемычку вызывает частичное испарение и окисление нихрома, что приводит к разрыву перемычки. Однако по истечении некоторого времени такая перемычка может восстановиться, поэтому для повышения надежности программирования проводят электротермотренировку микросхем. Подобного недостатка лишены микросхемы с перемычками из поликристаллического кремния, в которых процесс необратимого перехода поликремния из проводящего состояния в непроводящее происходит под воздействием нагрева, вызванного протеканием тока.

При возбуждении словарной линии будут активизироваться (переходить в состояние «Лог. 1») лишь те разрядные линии, к которым подсоединены транзисторы с невыплавленными перемычками. Таким образом, процесс программирования для микросхем данного типа сводится к удалению плавких перемычек в необходимых местах.

Схемы поддержки режима программирования обычно располагаются на самом кристалле микросхемы и процесс программирования протекает следующим образом. На адресные входы подается адрес выбранной ячейки. Напряжение питания микросхемы повышается до напряжения программирования, необходимого для создания тока, достаточного для выплавления перемычки. Далее на выходах микросхемы путем задания тока указываются те разряды слова, в которых будут выплавляться перемычки. В процессе занесения информации в микросхему необходимая последовательность подачи импульсов напряжения на определенные выводы обеспечивается программирующим устройством, которое параллельно контролирует правильность программирования, считывая информацию из ПЗУ. Ясно, что микросхемы ПЗУ данного типа допускают только однократную запись ячейки.

Микросхемы со стиранием информации с помощью ультрафиолетового излучения или ПЗУ с УФ-стиранием. Возможность многократного программирования, достаточно малые время выборки и энергопотребление, дешевизна, большая емкость делают их наиболее предпочтительными для применения в качестве памяти микропроцессорных систем с сохранением информации после отключения питания. Микросхемы данного типа используются в блоках ПЗУ большинства микроЭВМ.

Запоминающим элементом в ПЗУ с УФ-стиранием является МОП-транзистор, расположенный на пересечении соответствующих словарной и разрядной линий. Затвор этого транзистора подключен к словарной линии, сток — к $+u_{п}$, исток — к разрядной линии. Информация о содержимом данной ячейки хранится в виде заряда на втором, полностью изолированном (плавающем) затворе МОП-транзистора. Плавающим затвор называется потому, что он размещен между управляющим затвором данного транзистора и его каналом и окружен высокоомным диэлектриком. Такое положение затвора позволяет формировать вольт-амперную характеристику транзистора, изменяя напряжение отпирания. Если ввести в плавающий затвор электроны методом лавинной инжекции (туннелирования), то захваченный им отрицательный заряд будет формировать смещенную характеристику транзистора. Чем больше инжектировано электронов на плавающий затвор, тем сильнее смещается напряжение отпирания МОП-транзистора в сторону бóльших напряжений.

Процесс считывания из микросхем ПЗУ данного типа происходит таким образом: при подаче адреса на дешифратор некоторого адреса возбуждается соответствующая словарная линия, соединенная с затворами МОП-транзисторов, хранящих информацию. Если напряжение, приложенное к затвору (близкое к напряжению питания), превышает пороговое для данного транзистора, он переходит в проводящее состояние, т. е. на соответствующей разрядной линии будет «Лог.1». Если напряжение на затворе меньше порогового, то выбранный транзистор проводить не будет («Лог. 0»). Таким образом, осуществляя выборочную инжекцию электронов в плавающие затворы и изменяя пороговое напряжение транзисторов, можно заносить информацию в ПЗУ. При необходимости в перепрограммировании микросхемы предварительно записанную информацию стирают ультрафиолетовым светом через прозрачное кварцевое окошко на поверхности корпуса микросхемы. УФ-излучение, попадая на плавающий затвор и выбивая из него фотоэлектроны, разряжает плавающий затвор МОП-транзистора. Время сохранения информации в микросхемах ПЗУ данного типа определяется качеством призатворного диэлектрика и составляет для современных микросхем десять и более лет.

В соответствии с техническими условиями для программирования таких микросхем ПЗУ требуется подача высокого напряжения (+24 В, +21 В или +12 В) на специальный вывод микросхемы в течение 50 мс. За это время ускоренные промежутком «управляющий затвор — канал» электроны туннелируют в плавающий затвор и захватываются им. Для микросхем емкостью 1К×8 время записи во все ячейки составляет около 1,5 мин. При записи в микросхемы емкостью 4К×8, 8К×8,

32К×8 и т. д. это время значительно возрастает. Однако исследования показали, что инжектируемый на плавающий затвор заряд в этом случае часто бывает избыточным, так как 90 % всех ячеек программируются уже за время около 5 мс.

Уменьшить время программирования микросхем емкостью более 1К×8 можно, воспользовавшись алгоритмом Inteligent Algorithm, разработанным фирмой INTEL или High Performance Programming Algorithm, предложенным фирмой Hitachi [1—5]. Суть его заключается в том, что программирование производится короткими импульсами (1 мс) с проверкой ячейки после каждого импульса. При переходе ячейки из незапрограммированного состояния в запрограммированное (из единицы в ноль) подается фиксирующий импульс, равный по длительности утроенному времени программирования данной ячейки. Если после подачи 25 программирующих импульсов ячейка оказывается незапрограммированной, процесс прерывается и выдается сообщение об ошибке (рис. 1).

Второй важный момент состоит в подаче на микросхему во время программирования и проверки напряжения питания величиной не +5 В, как обычно, а +6 В. Это связано с

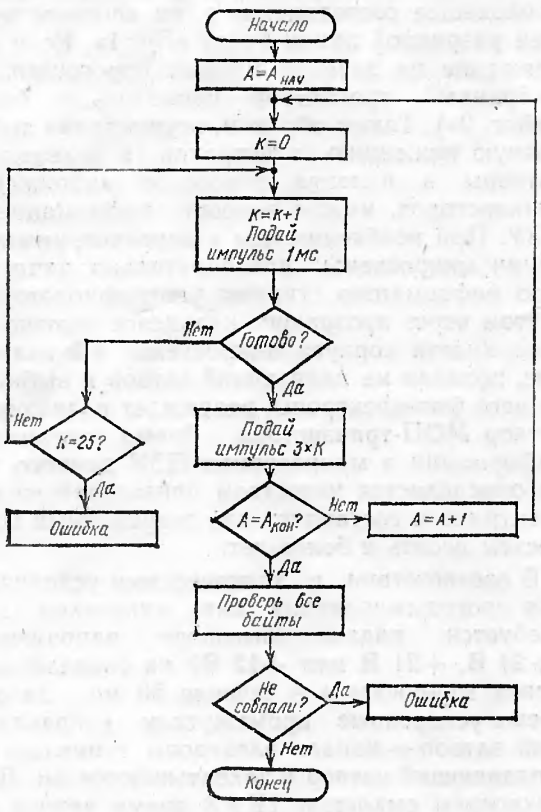
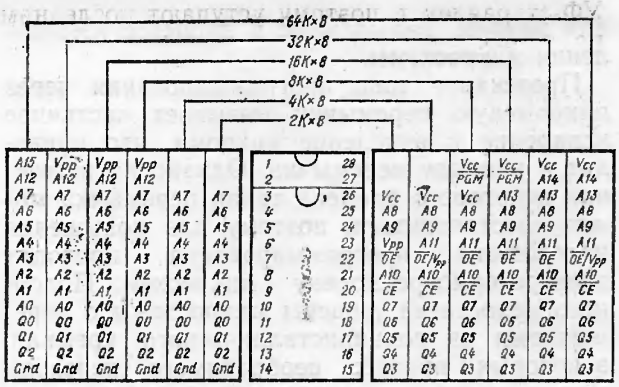


Рис. 1. Алгоритм ускоренной записи информации в ПЗУ с УФ-стиранием



2К×8		4К×8		8К×8		16К×8		32К×8		Номер выводов
Запись	Чтение	Запись	Чтение	Запись	Чтение	Запись	Чтение	Запись	Чтение	
–	–	–	–	Vpp	+5В	Vpp	+5В	Vpp	+5В	1
⎍	0	⎍	0	0	0	0	0	0	0	20
⎍	0	Vpp	0	1	0	1	0	⎍	0	22
Vpp	+5В	A11	A11	A11	A11	A11	A11	A11	A11	23
+5В	+5В	+5В	+5В	–	–	A13	A13	A13	A13	26
–	–	–	–	⎍	?	⎍	1	A14	A14	27

Рис. 2. Цоколевка выводов и комбинации сигналов записи, проверки и чтения информации для микросхем ПЗУ с УФ-стиранием

тем, что если некоторая ячейка памяти (МОП-транзистор) будет иметь пороговое напряжение открывания, например +5,5 В, то при работе микросхемы в реальных условиях с вариациями питающего напряжения может случиться так, что запрограммированный транзистор при превышении напряжения питания +5,5 В откроется и будет вести себя как незапрограммированный, приводя к неверному считыванию информации из микросхемы. Чтобы этого избежать, проверка ячейки ведется при повышенном напряжении питания, гарантируя тем самым работу в промышленном диапазоне питающих напряжений без сбоев. Использование указанного алгоритма уменьшает время программирования микросхем в 5—10 раз при сохранении надежности.

Для того чтобы программировать микросхемы различной емкости, программатор должен иметь стандартный 28-выводной разъем. Цоколевка микросхем ПЗУ с УФ-стиранием емкостью 4К, 8К, 16К, 32К байт, принятая решением комиссии JEDEC в качестве стандарта, и комбинации сигналов записи проверки и чтения информации для микросхем ПЗУ с УФ-стиранием приведены на рис. 2.

Микросхемы ПЗУ с электрическим стиранием информации привлекают разработчиков микропроцессорной техники возможностью быстрого стирания и записи, большим допустимым числом циклов перезаписи информации (10 000 раз и более). В настоящее время они достаточно дороги и сложны по сравнению с микросхемами ПЗУ с

УФ-стиранием и поэтому уступают последним по степени использования в МП-аппаратуре.

Основу запоминающей ячейки в ПЗУ с электрическим стиранием составляет МОП-транзистор с плавающим затвором, такой же как и в ПЗУ с УФ-стиранием. Но в микросхемах данного типа технологическими методами обеспечена возможность обратного туннелирования, т. е. отбора электронов с плавающего затвора, что позволяет выборочно стирать занесенную информацию. В связи со сходством запоминающих элементов, все приведенное выше для микросхем с УФ-стиранием справедливо и для микросхем с электрическим стиранием.

Для создания специальных инструментальных средств занесения информации в ПЗУ — программаторов — необходимо знание алгоритмов записи, обзору которых посвящена следующая статья цикла.

ЛИТЕРАТУРА

1. Van Buskirk M. et. al.— Electronics, Feb., 1983.
2. Knowlton D. Versatile algorithm, equipment cut EPROM programming time.— EDN, March, 1983, p. 175.
3. INTEL Memory Components Handbook, 1984.
4. HITACHI IC Memory Data Book, 1984.
5. Экспресс-информация. Вычислительная техника, № 47, 1984 г. с. 1—7.

Статья поступила 15 мая 1985 г.

УДК 681.326.3

А. П. Дигнов, Н. Н. Щелкунов

МЕТОДИКА ПРОГРАММИРОВАНИЯ МИКРОСХЕМ ПЗУ

Микросхемы типа К556РТ4

Для записи информации в микросхему на ряд ее выводов подаются импульсы напряжения определенной амплитуды и длительности (рис. 1). До программирования по всем адресам и разрядам в микросхеме содержится состояние «Лог. 0».

В каждом цикле программируется только один разряд выбранного слова. В исходном состоянии напряжение на выводах 13, 14 и 16 равно нулю. Запись бита информации выполняется в следующем порядке:

на адресные входы подается код адреса слова, в котором программируется бит;

напряжение питания повышается до 5±±0,25 В, источник напряжения при этом должен обеспечить ток не менее 150 мА;

на выводы 9...12 подается напряжение «Лог. 0»;

напряжение питания повышается до 12,5±±0,5 В, источник должен обеспечить ток не

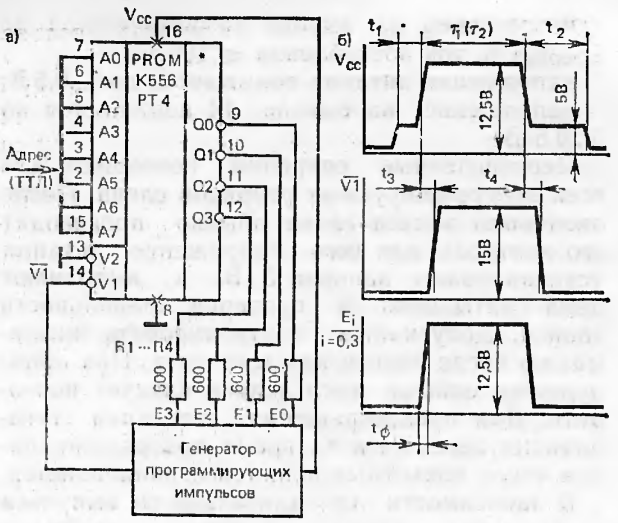


Рис. 1. Схема подключения ПЗУ К556РТ4 в режиме записи информации (*а*), временные диаграммы (*б*)

менее 400 мА; такое же напряжение E_i, $i=0...3$, через резистор $R=600$ Ом подается на выход, соответствующий программируемому разряду, ток потребления $\leqslant 15$ мА;

Таблица 1

Параметры программирования микросхем ПЗУ типа К556РТ4

Длительность импульсов программирования, мкс (мс)	$25 \leqslant \tau_1 \leqslant 100$ ($10 \leqslant \tau_2 \leqslant 15$)
Число импульсов программирования на один бит	$1000 \leqslant N_1 \leqslant 4000$, $N_2 = 100$, $40 \leqslant N_3 \leqslant 100$
Скважность программирующих импульсов	$10 \leqslant Q \leqslant 20$
Время предустановки напряжения питания	200 нс $\leqslant t_1 \leqslant 10$ мкс
Время воздействия напряжения питания при контроле, мкс	$1 < t_2 \leqslant 30$
Время задержки подачи импульсов на входе разрешения выборки ($\overline{V2}$), мс	$0 \leqslant t_3 \leqslant 100$
Время задержки снятия импульсов на входе разрешения выборки ($\overline{V2}$), мс	$200 \leqslant t_4 \leqslant 1000$
Длительность фронта и спада импульса, нс	$300 \leqslant t_\phi \leqslant 500$

напряжение на выводе 14 повышается до 15±0,5 В, ток потребления ⩽100 мА;

напряжение питания понижается до 0...0,5 В;

напряжение на выводе 14 понижается до 0...0,5 В.

Перечисленные операции повторяют для всех программируемых разрядов слова. После окончания записи слова обычно производят его контроль, для чего напряжение питания устанавливают равным 5 В и выполняют цикл считывания и проверки правильности записи. Допускается контролировать информацию после записи каждого бита. При обнаружении ошибок цикл записи следует повторить. Для предотвращения зависания тугоплавких перемычек во время программирования число повторных попыток ограничивается.

В зависимости от длительности импульса напряжения питания τ и числа допустимых попыток N различают три режима программирования ИС: нормальный (τ_1, N_1), форсированный (τ_2, N_2) и дополнительный (τ_1, N_3) (табл. 1).

Программирование слова начинают в нормальном режиме. Если нормальный режим не обеспечил правильную запись информации, то переходят к форсированному режиму. В случае неудачи и в этом режиме микросхема считается негодной. При первом успешном прохождении операции контроля, независимо от того, в нормальном или форсированном режимах программирования он был получен, необходимо перейти в дополнительный режим. Программирование следующего слова начинают после успешного выполнения всего дополнительного режима. При обнаружении ошибки контроля в дополнительном режиме (восстановление перемычки) делается повторная попытка записи информации.

Микросхемы с записанной информацией рекомендуется подвергать электротермотренировке со считыванием информации с частотой 50 Гц...1 МГц последовательно по всем адресам микросхемы. Электротермотренировка проводится в течение 168 ч при температуре среды 70 °С. Микросхемы, у которых в процессе тренировки произошла потеря записанной информации, допускается программировать один раз повторно.

Микросхемы типа К541РТ1

Микросхема ПЗУ К541РТ1 по информационной емкости и назначению выводов аналогична микросхеме К556РТ4. Однако микросхема К541РТ1 не требует электротренировки, поскольку использование в ней поликремниевых перемычек исключает возможность их восстановления после программирования (см. предыдущую статью).

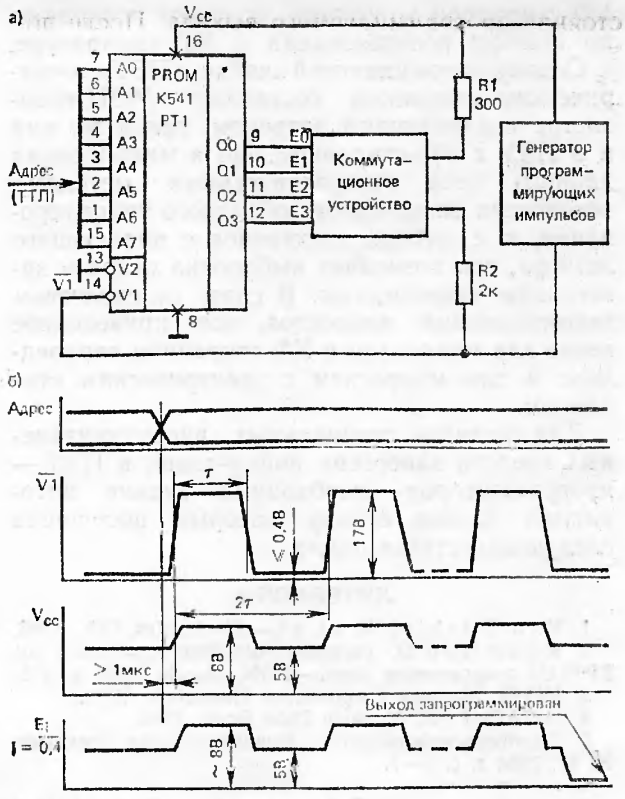


Рис. 2. Схема подключения ПЗУ К541РТ1 в режиме записи информации (*а*), временные диаграммы (*б*)

До программирования в ПЗУ по всем адресам и разрядам записана «Лог. 1». Структурная схема подключения К541РТ1 и временные диаграммы программирования приведены на рис. 2.

В исходном состоянии на вывод *16* подается напряжение питания 5±0,5 В. Выход, подлежащий программированию, через резистор *R1* присоединен к выводу *16*. Резистор *R2* отключен от выхода. На всех остальных выходах должно быть установлено напряжение «Лог. 0».

Запись бита информации выполняется в следующем порядке:

на адресные входы подается код слова, в котором программируется бит;

напряжение на выводе *14* повышается до 17±0,5 В, максимальный программирующий ток 250 мА;

напряжение на выходе *16* повышается до 8±0,5 В, ток потребления ⩽200 мА;

на программируемый выход через *R2* подается ток программирования <4 мА, резистор *R1* отключается от выхода.

Длительность импульсов, подаваемых на вывод *14*, линейно возрастает от 1 до 8 мкс в течение 400 мс при скважности, равной 2 В паузах между импульсами контролируется со-

стояние программируемого выхода. После появления на выходе напряжения низкого уровня цикл программирования продолжается в течение 100 мс.

Состояние выхода в цикле программирования можно не контролировать. При этом время цикла устанавливается 400 мс, а напряжение на выводе *16* в течение всего цикла поддерживается равным 8±0,5 В. При условии записи бита, но не ранее чем через 50 мс, переходят к записи следующего бита.

Микросхемы типа К556РТ5

Схема подключения микросхемы К556РТ5 при записи информации приведена на рис. 3. До программирования по всем адресам и разрядам микросхемы записан высокий логический уровень. В каждом цикле программируется только один бит.

В исходном состоянии на выводы *21, 22, 24* подается напряжение «Лог. 0», а на выводы *18, 19* — «Лог. 1». Для записи бита информации выполняют следующую последовательность операций:

на адресные входы подается код адреса слова;

напряжение питания повышается до 5±±0,25 В, источник напряжения должен обеспечить ток не менее 200 мА;

на выводы *9...11, 13...17* подается напряжение «Лог. 0»;

напряжение питания повышается до 12,5±±0,5 В, ток, обеспечиваемый источником в этой фазе программирования, не менее 600 мА;

такое же напряжение через резистор 300 Ом подается на программируемый разряд выходной шины; ток нагрузки ⩽15 мА;

напряжение на выводе *21* повышается до 15±0,5 В, ток потребления ⩽100 мА;

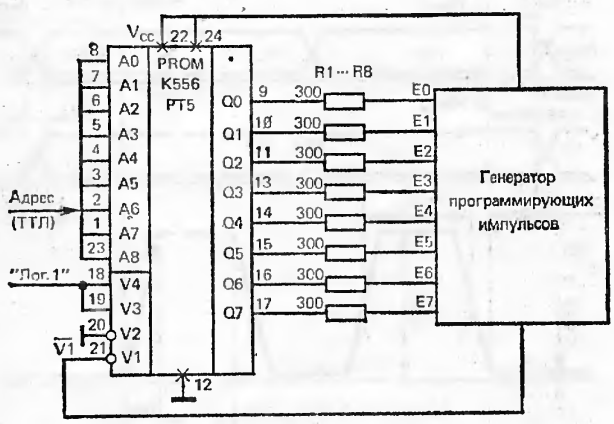


Рис. 3. Схема подключения ПЗУ К556РТ5 в режиме записи информации

напряжение питания понижается до 0...0,5 В;

напряжение на выводе *21* понижается до 0...0,5 В.

Процесс записи информации в микросхемы типа К556РТ5 аналогичен процессу программирования приборов К556РТ4 с теми же параметрами трех возможных режимов (см. табл. 1).

Микросхемы типа К556РТ7

Схема подключения микросхем ПЗУ типа К556РТ7 при записи информации приведена на рис. 4,*а*. До программирования в микросхемах по всем адресам и разрядам записан низкий логический уровень «Лог. 0». В исходном состоянии на выводы *20, 24* подается напряжение «Лог. 0», а на выводы *18, 19* — «Лог. 1». Последовательность подачи напря-

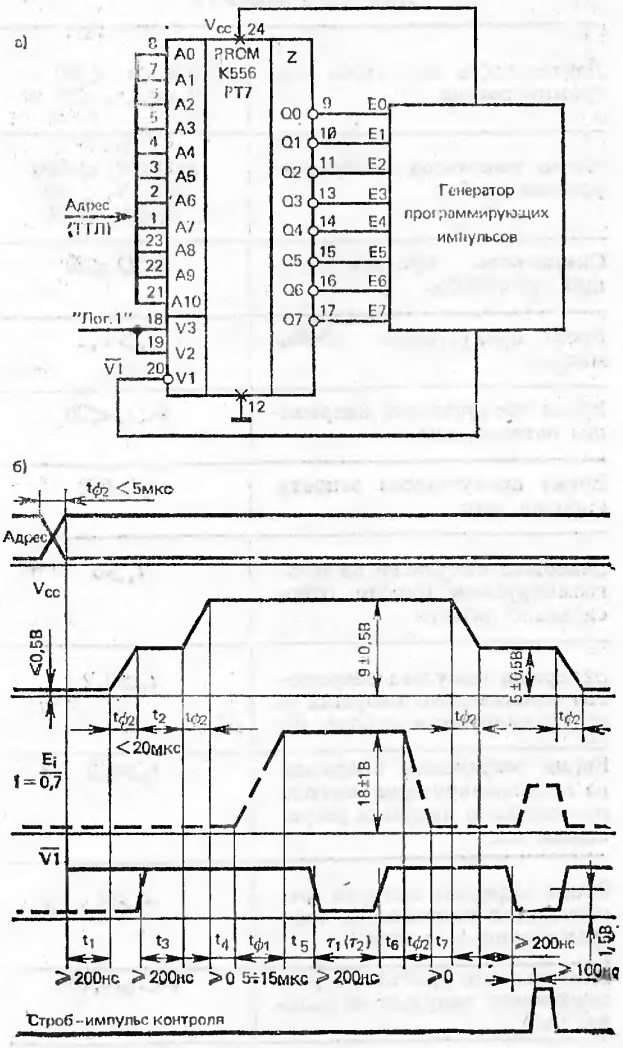


Рис. 4. Схема подключения ПЗУ К556РТ7 в режиме записи информации (*а*), временные диаграммы (*б*)

жений при программировании приведена на рис. 4,*б* и заключается в следующем:

на адресные входы подается код адреса;

напряжение питания (вывод *24*) повышается до 5±0,5 В, источник должен обеспечить ток не менее 200 мА;

на вывод *20* подается напряжение «Лог. 1», при этом все выходы переходят в высокоимпедансное состояние;

напряжение питания повышается до 9± ±0,5 В, источник должен обеспечить ток ⩾500 мА;

на программируемом выходе напряжение повышается до 18±1 В, источник должен обеспечить ток ⩾150 мА;

напряжение на выводе *20* понижается до 0...0,5 В (это состояние микросхемы соответствует записи);

напряжение на выводе *20* повышается до «Лог. 1»;

с программируемого разряда снимается напряжение 18 В;

напряжение питания понижается до 0...0,5 В.

Полный цикл программирования производится аналогично записи информации в ПЗУ К556РТ4. Параметры режимов программирования: нормального (τ_1, N_1), форсированного (τ_2, N_2), дополнительного (τ_1, N_3), а также ряд длительностей временной диаграммы (см. рис. 4) приведены в табл. 2.

Таблица 2

Параметры программирования микросхем ПЗУ типа К556РТ7

Длительность импульсов программирования	25 мкс $\leqslant\tau_1\leqslant$100 мкс 10 мс $\leqslant\tau_2\leqslant$25 мс
Число импульсов программирования на бит	$1000\leqslant N_1\leqslant 4000$ $40\leqslant N_2\leqslant 100$ $40\leqslant N_3\leqslant 100$
Скважность программирующих импульсов	$5\leqslant Q\leqslant 30$
Время предустанова адреса, мкс	$t_1\geqslant 0{,}2$
Время предустанова напряжения питания, мкс	$0\leqslant t_2\leqslant 20$
Время предустанова запрета выборки, мкс	$t_3\geqslant 0{,}2$
Задержка импульсов на программируемом выходе относительно питания	$t_4\geqslant 0$
Задержка импульса разрешения относительно импульса на программируемом выходе, мкс	$t_5\geqslant 0{,}2$
Время сохранения импульса на программируемом выходе относительно импульса разрешения, мкс	$t_6\geqslant 0{,}2$
Время задержки питания относительно импульса на программируемом выходе	$t_7\geqslant 0$
Длительность фронта программирующего импульса на выходе, мкс	$5\leqslant t_{\phi 1}\leqslant 15$
Фронты и спады импульсов на входах (выходах), мкс	$t_{\phi 2}\leqslant 5$

Микросхемы типа К573РФ1

Первоначально, а также после стирания информации по всем адресам и разрядам микросхем, записан высокий логический уровень

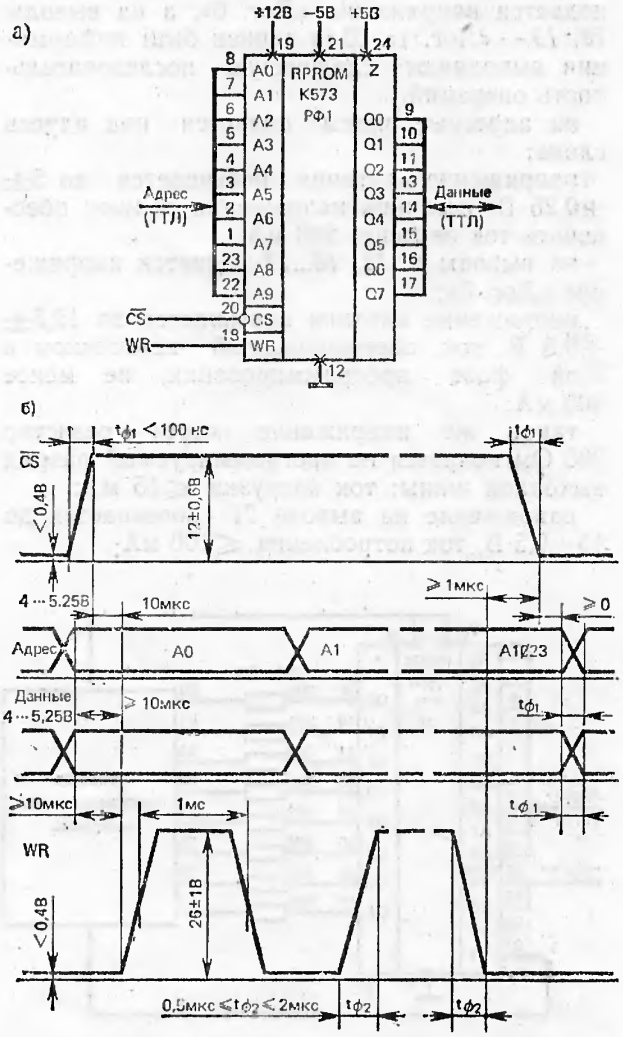


Рис. 5. Режим записи информации в ПЗУ К573РФ1: схема подключения (*а*), временные диаграммы (*б*)

напряжения — «Лог. 1». Программирование состоит в записи напряжения «Лог. 0» в выбранную ячейку матрицы ПЗУ. Запись информации в микросхему производится в соответствии с временной диаграммой, приведенной на рис. 5, при обязательном последовательном переборе всех адресов. В каждом шаге программирования восемь разрядов записываемого кода в виде ТТЛ-уровней напряжения поступают на выходные линии *Q0...Q7* микросхемы и стробируются импульсом записи WR, длительностью $t_{прог}=0{,}1...1$ мс. Затем значение адреса инкрементируется, и аналогично программируется следующая ячейка. Перебор всех 1024 адресов образует один цикл занесения информации. Для надежной записи необходимо от 100 (при $t_{прог}=1$ мс) до 1000 (при $t_{прог}=0{,}1$ мс) таких циклов.

По выводу *18* (вход сигнала записи) программатор необходимо рассчитывать на нагрузку до 30 мА при емкости 100 пФ. Источник напряжения питания — 5 В должен включаться первым, а выключаться последним. Допускается одновременное включение и выключение источников питающих напряжений.

Стереть записанную информацию можно УФ-излучением с длиной волны $\lambda \leqslant 400$ нм. В процессе стирания выводы микросхемы должны быть замкнуты между собой. Рекомендуется интегральная доза облучения 10 Вт·с/см² при энергетической освещенности 100 Вт/м².

Микросхемы К573РФ2 и К573РФ5

Первоначально, а также после стирания информации по всем адресам и разрядам микросхем, записан высокий логический уровень напряжения — «Лог. 1». Информация стирается так же, как в микросхеме К573РФ1.

При подаче на вывод *21* напряжения +26 В микросхема переходит в режим программирования. После установления кода адреса и кода записываемого числа на вход WR подается высокий ТТЛ-уровень. Цикл записи состоит в генерации одного импульса программирования на каждое слово (рис. 6). Затем адрес инкрементируется и цикл повторяется. Для уверенной записи информации необходимо 250 повторов записи по каждому адресу микросхемы. За режимом записи следует операция проверки содержимого БИС (при этом допустимо не снимать напряжение +26 В с вывода *21*). При нормальной эксплуатации микросхем напряжение на этом выводе должно быть равно напряжению питания.

Допускается программирование микросхем К573РФ2, К573РФ5 однократным импульсом

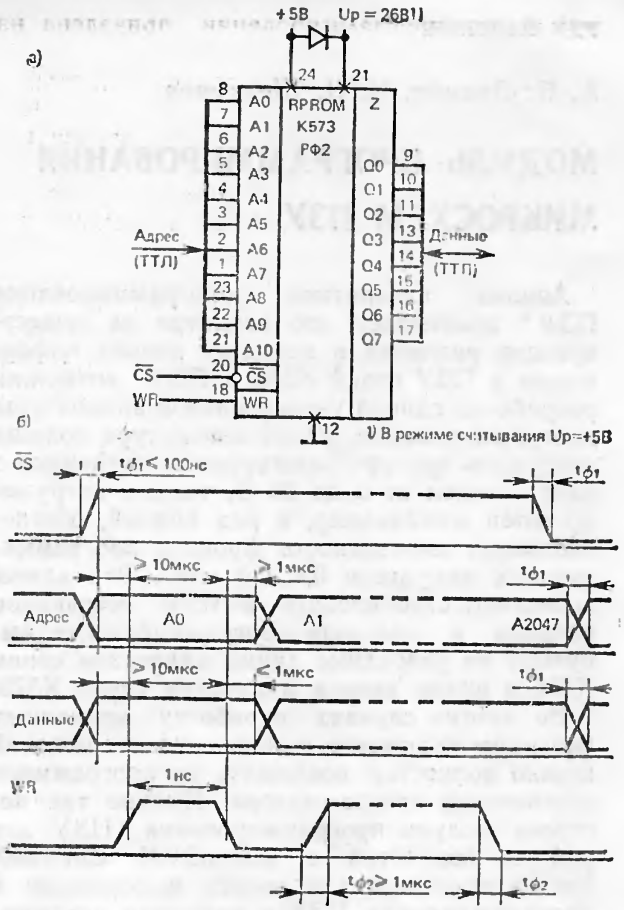


Рис. 6. Режим записи информации в ПЗУ типа К573РФ2 и РФ5:
Схема подключения (*а*), временные диаграммы (*б*)

длительностью ~50 мс, причем можно программировать ячейки в любом порядке и без обязательного перебора всех адресов. В частности, можно изменять содержимое частично запрограммированных микросхем. При этом возможно только изменение состояния бит с «Лог. 1» на «Лог. 0».

ЛИТЕРАТУРА

1. Микросхемы интегральные. Руководство по применению. ОСТ 11 340.913—81.

2. Микросхемы интегральные. Руководство по применению. ОСТ 11 340.916—82.

3. Полупроводниковые запоминающие устройства и их применение / В. П. Андреев, В. В. Баранов, Н. В. Бекин и др.; Под ред. А. Ю. Гордонова.— М.: Радио и связь, 1981.—344 с.

4. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах: Программирование, типовые решения, методы отладки.— М.: Радио и связь, 1984.— 270 с.

Статья поступила 29 марта 1985 г.

УДК 681.322.068

А. П. Дианов, Н. Н. Щелкунов

МОДУЛЬ ПРОГРАММИРОВАНИЯ МИКРОСХЕМ ПЗУ

Анализ алгоритмов программирования ПЗУ* показывает, что несмотря на существующие различия в методике записи информации в ПЗУ серий К556 и К573 возможна разработка единой универсальной аппаратуры программирования. Такая аппаратура должна содержать три программируемых источника с напряжением от 0 до 26 В, токами нагрузки до сотен миллиампер, и ряд ключей, обеспечивающих длительность фронтов программирующих импульсов 0,4...0,6 мкс. Эти ключи позволяют стробировать выходы источников питания и подавать программирующие импульсы на разрядные линии микросхем серии К556 и входы записи микросхем серии К573.

Во многих случаях отработку временных диаграмм включения-выключения напряжений можно полностью возложить на программное обеспечение программатора. Именно так построен модуль программирования ПЗУ для работы совместно с микроЭВМ СМ-1800, предназначенный для записи информации в программируемые ПЗУ с плавкими перемычками (серия К556) и УФ-стиранием (серия К573).

При разработке программатора учтены условия его совместимости с универсальной программой прожига UPP версии 3.7. Эта оверлейная программа общим объемом в 43,5 Кбайт входит в состав тестового обеспечения операционной системы СМ-1800 и предназначена для управления процессом прожига ПЗУ серии К556. Встроенный в программу диалоговый монитор обеспечивает прием и исполнение 17 различных команд, реализующих ввод, редактирование, прожиг, контроль и документирование содержимого четырех типов ПЗУ. Для занесения информации в УФРПЗУ серии К573 написаны подпрограммы, составляющие с программой UPP единый пакет программирующих программ. Пакет обладает необходимым набором сервисных средств для работы с ПЗУ.

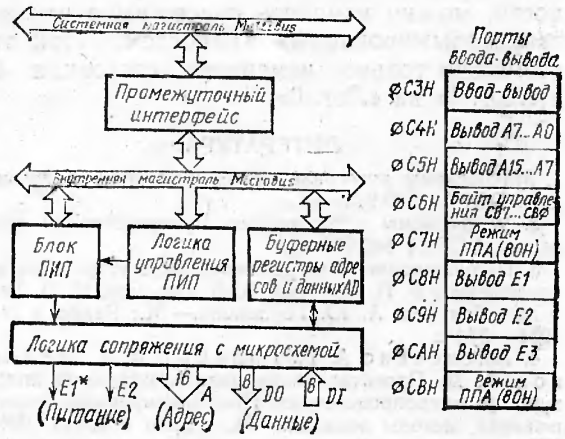


Рис. 1. Структурная схема модуля программирования

Модуль программирования ПЗУ (рис. 1) содержит промежуточный интерфейс И41 (Multibus) — Microbus, блок программируемых источников питания (ПИП), логику управления ПИП, буферные регистры адресов и данных, логику сопряжения с микросхемой.

Модуль подключается к одной из широко распространенных системных магистралей (Multibus). На ее основе построено множество микросистем, в том числе микроЭВМ СМ-1800. При изменении промежуточного интерфейса модуль можно подключить к шинам других ЭВМ, например к МПИ «Электроника 60» или к «Общей шине» ЭВМ «Электроника 100/25». Для микроЭВМ модуль представляет собой некоторый набор портов ввода-вывода, обращение к которым инициируется программой (см. рис. 1).

Для сопряжения с микросхемой используется 35 линий:

стробируемые линии E1* и E2 с программируемой амплитудой;

адресная шина A15...A0;

шина считанных данных D17...D10;

стробируемая шина данных записи D07...D00 с программируемой амплитудой E3;

общая линия GND.

Во всех случаях длительность стробирования задается программно.

Промежуточный интерфейс

Промежуточный интерфейс программатора осуществляет сопряжение внутренней магистрали Microbus с системной шиной Multibus (рис. 2). Она содержит: шинный формирователь DD1, DD2, дешифратор адреса DD3 для получения сигналов выборки $\overline{CSD}$, $\overline{CS1}$, $\overline{CS2}$ и CSM, буферные схемы сигналов управления $\overline{I/OR}$, $\overline{I/OW}$ и адреса AB0, AB1, а также формирователь сигнала подтверждения обмена XACK DD6.1, DD7 и DD8.1. Внутренняя шина Microbus при этом работает в синхронном режиме, а длительность стробов $\overline{I/OR}$ и $\overline{I/OW}$ определяется задержкой XACK относительно сигналов $\overline{I/ORC}$ или $\overline{I/OWC}$.

* А. П. Дианов, Н. Н. Щелкунов. Методика записи информации в ПЗУ (см. предыдущую статью на с. 75).

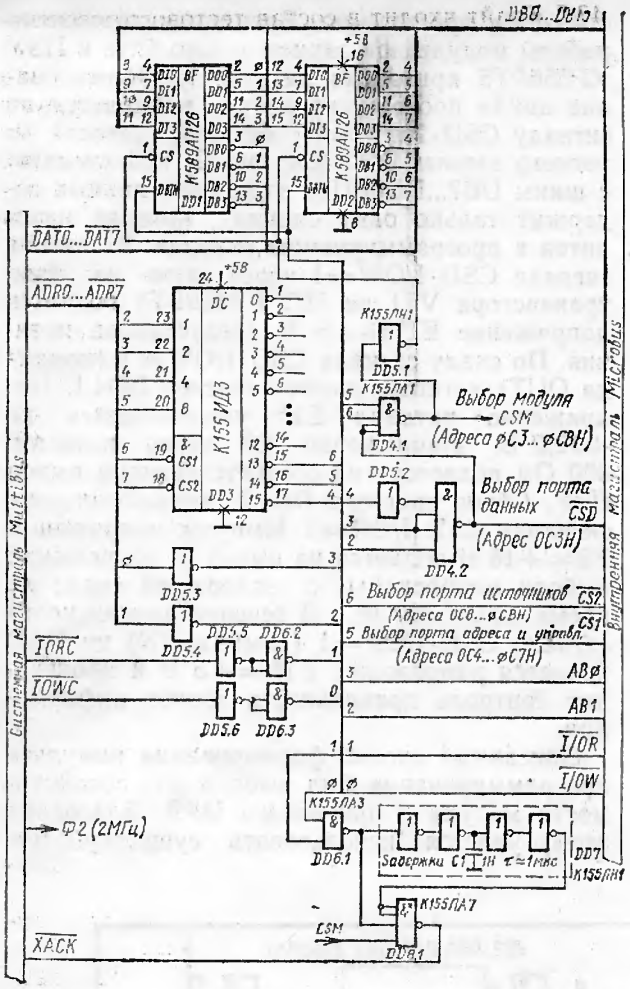


Рис. 2. Схема промежуточного интерфейса

Блок программируемых источников питания

При записи информации в микросхемы необходимо иметь напряжения, регулируемые в пределах 5...26 В с током нагрузки до сотен миллиампер. В рассматриваемом модуле имеются три программируемых источника питания (ПИП) Е1, Е2 и Е3 (рис. 3). Каждый источник содержит 8-разрядный буферный регистр, в качестве которого используется один из портов программируемого периферийного адаптера (ППА) КР580ВВ55, цифро-аналоговый преобразователь К572ПА1, быстродействующий операционный усилитель К544УД2 и выходные транзисторы КТ644 и КТ909. Максимальная скорость нарастания выходного напряжения ПИП составляет ~20 В/мкс. При записи в ПЗУ типа КМ556РТ7 эта скорость оказывается слишком большой, поэтому предусмотрена возможность ее уменьшения в ПИП Е3. Для этого на вход «+» его операционного

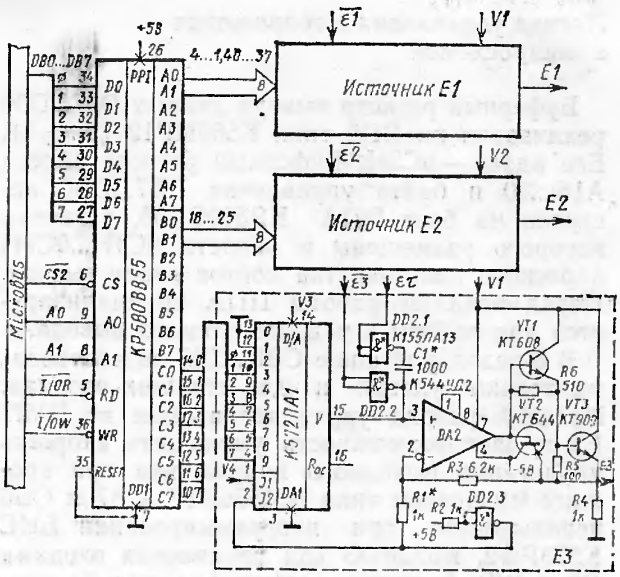


Рис. 3. Электрическая схема программируемых источников питания

усилителя ОУ при εт, равном «Лог. 1», подключается дополнительный конденсатор *С1*. Включение источников производится сигналами ε1, ε2 и ε3 соответственно. В схеме источника использовано нетрадиционное включение ЦАП К572ПА1, в котором полярность выходного напряжения ПИП и опорного напряжения ЦАП совпадают.

ППА блока источников питания инициализируется для работы в режиме прямого вывода (режим 0) по всем трем каналам *A*, *B* и *C*, управляющим словом 80Н. Регистры ППА расположены в зоне С8Н...СВН адресного пространства ввода-вывода микросистемы (см. рис. 1). Шаг изменения напряжения равен 0,1 В. Диапазон регулировки выходных напряжений источников *Е1* и *Е3* лежит в пределах 0...18 В, а источника *Е2* — 0...25,6 В.

Для получения необходимых напряжений питания V1...V4 цифро-аналоговых преобразователей и операционного усилителя блок ПИП имеет встроенный преобразователь напряжения (рис. 4). Модуль питается от двух источников напряжением +5 и +12 В.

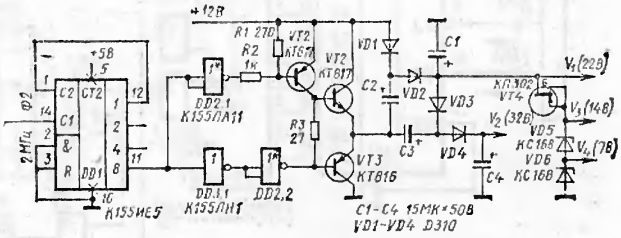


Рис. 4. Электрическая схема преобразователя напряжения

Логика управления и сопряжения с микросхемой

Буферный регистр вывода данных D07...D00 реализован на DD5 типа К589ИР12 (рис. 5). Его адрес — 0С3Н. Буферный регистр адреса A15...A0 и байта управления СВ7...СВ0 построен на базе ППА КР580ВВ55, регистры которого размещены в области 0С4Н...0С7Н адресного пространства портов ввода-вывода. Перед началом работы ППА инициализируется для работы в режиме прямого вывода.

В 8-разрядной шине СВ0...СВ7 задействованы только нулевой и два старших разряда. Нулевой разряд управляет входом εт ПИП Е3 и дает возможность уменьшить скорость нарастания выходного напряжения при прожиге микросхем типа КМ556РТ7. СВ7 и СВ6 используются при программировании БИС К573РФ2, К573РФ5 для управления входами CS и WR. В состав модуля (см. рис. 5) входят ключи на транзисторах VT1...VT9 и формирователь программирующих напряжений на триггере DD4.1. Подключение микросхем ПЗУ различных типов к модулю программирования показано на рис. 6.

Управление величиной и длительностью программирующих напряжений, формирование импульсов программирования и контроль правильности записанной информации осуществляется программно. Временна́я диаграмма работы модуля при записи одного бита в ПЗУ КР556РТ5 приведена на рис. 7. Формирование цикла программирования начинается по сигналу CSD·I/OW=1, который заносит в регистр данных DD5 (см. рис. 5) информацию с шины DB7...DB0. При этом байт данных содержит только одну единицу, которая находится в программируемом разряде. В течение сигнала CSD·I/OW=1 через ключ на базе транзистора VT1 на ИС КР556РТ5 подается напряжение Е1*=+5 В предустанова питания. По спаду сигнала CSD·I/OW=1 (команда OUT) устанавливается триггер DD4.1. Напряжение питания Е1* увеличивается до +12,5 В. Напряжение Е3 через резистор 300 Ом подается на соответствующий выход ПЗУ. Сброс триггера DD4.1 осуществляется сигналом CSD·I/OR=1. Импульс напряжения Е2=+15 В подается на вывод 21 разрешения выбора микросхемы с задержкой спада на время около 300 нс. В течение длительности сигнала CSD·I/OR=1 (команда IN) на ПЗУ подается напряжение Е1*=+5 В и происходит контроль правильности записи информации.

Описанный способ формирования импульса программирования был выбран для совместимости модуля и программы UPP. Благодаря этому удается использовать существующую

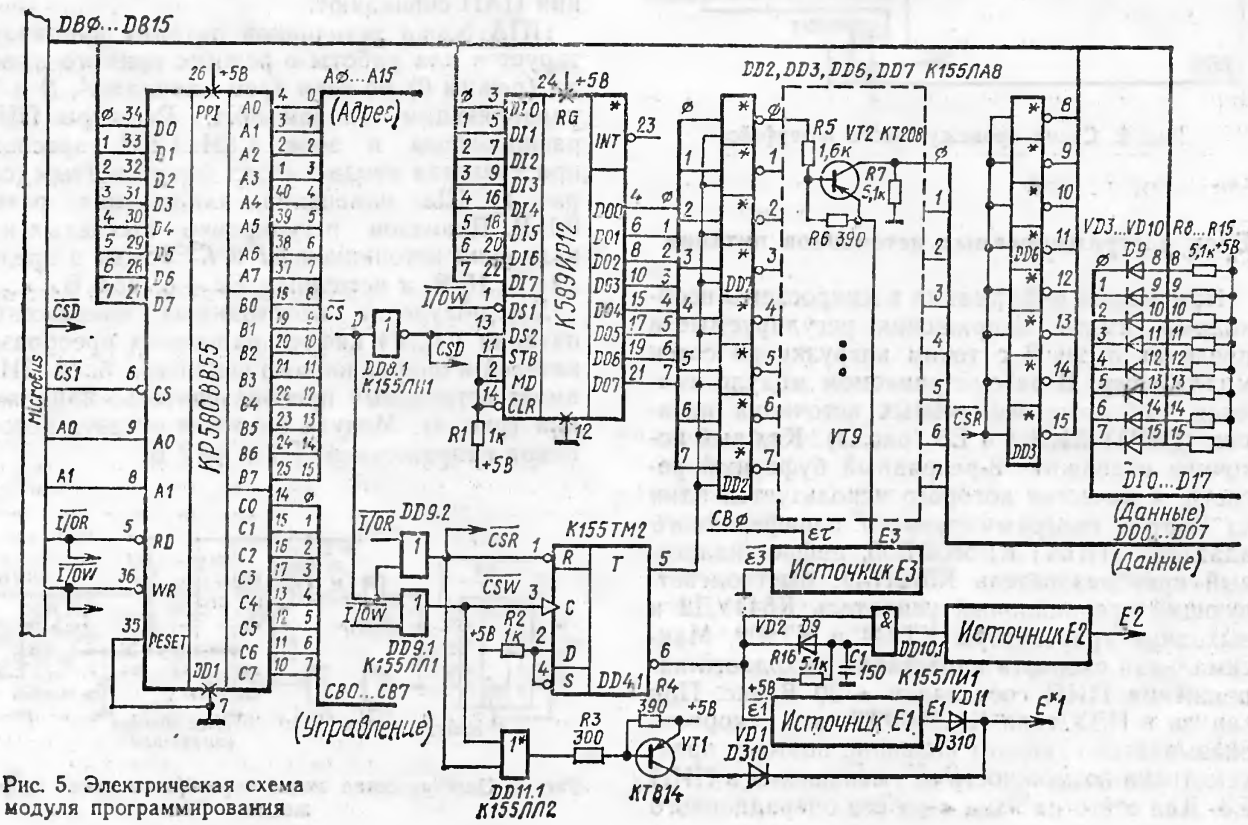


Рис. 5. Электрическая схема модуля программирования

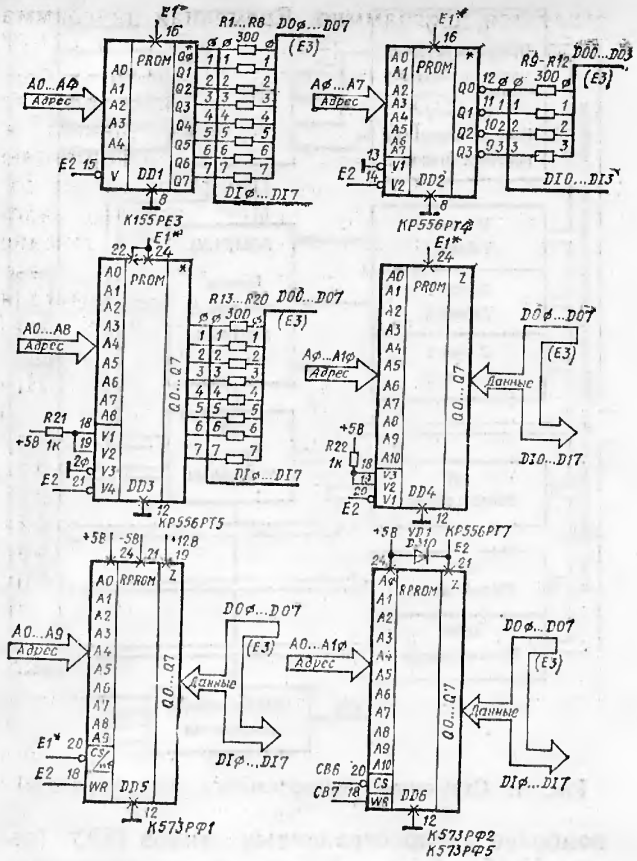


Рис. 6. Схемы подключения ПЗУ различных типов к модулю программирования

версию 3.7 UPP для прожига микросхем типа КР556РТ4 и КР556РТ5 без ее модификации.

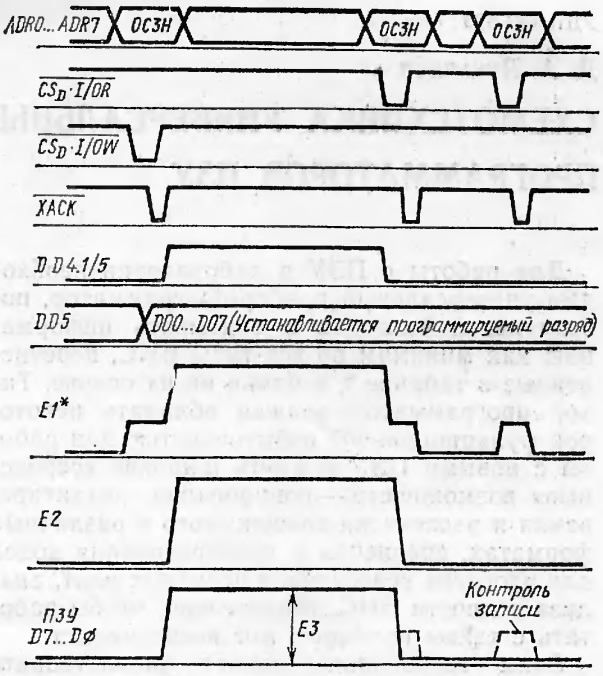


Рис. 7. Временна́я диаграмма записи одного бита информации в ЭРПЗУ КР556РТ5

Временные диаграммы записи информации в микросхемы ЭРПЗУ КМ556РТ7 и УФРПЗУ серии К573 формируются программным способом при $\varepsilon_i = 0$, $i = \overline{1,3}$. Для этого используется операция загрузки буферных регистров ПИП кодом, соответствующим величине напряжения.

Статья поступила 9 апреля 1985 г.

БИНТИ № 2 (2195)

Средства синтезирования и распознавания речи

Синтезаторы речи выпускаются сейчас несколькими фирмами и продаются по цене 5—10 фунтов стерлингов. Цена устройств преобразования текстов в речевую форму, выпускаемых фирмами «Диджитал экуипмент корпорейшн» и «Инфовокс», составляет 2000 фунтов стерлингов, а устройств распознавания речи — от 2 до 35 тыс. фунтов стерлингов в зависимости от состава словаря и числа распознаваемых голосов. Так, устройство распознавания речи «Логос», разработанное фирмой «Лоджика», первоначально имело словарь из нескольких сотен слов и распознавало один голос, а сейчас созданы его варианты для распознавания нескольких голосов, имеющие расширенный словарный запас.

По оценке фирмы «Техас инструментс», к 1988 году средства распознавания и синтезирования будут применяться в телефонной связи и устройствах регистрации речи, а также в бытовой технике, автомобилях, средствах управления производственными процессами, а в дальнейшем — в учрежденческой технике для непосредственного преобразования речи в тексты.

БИНТИ № 8 (2201)

Компьютер для слепых

Филипп Балин (авиатранспортная компания «Эр Франс») сконструировал компьютеризированную систему обработки слов по системе Брайля, используемой слепыми.

Эту систему можно подключать к обычным ЭВМ и обеспечивать слепым возможность работы с таким же электронно-вычислительным оборудованием, с которым работают зрячие. Пользоваться такой системой могут зрячие и слепые, не знающие азбуки Брайля или обычной письменности.

В систему Балина входят ЭВМ «Леонард», небольшой клавишный пульт Брайля и имитатор, заменивший обычный видеоэкран.

БИНТИ № 4 (2197)

Средства регистрации информации

Фирма «Шугарт» разработала ЗУ типа «Винчестер», емкость которого с двумя магнитными дисками диаметром 135 мм составляет 20 Мбайт.

Применение в этом ЗУ замкнутой системы сервоуправления позволило повысить плотность размещения до 285 магнитных дорожек на сантиметр. Новое ЗУ отличается также повышенной вибропрочностью; его размеры составляют 41×135×200 мм, а цена — 500 долл.

Вертикальные методы записи позволяют получить в ЗУ на магнитных дисках плотность записи 23500 битов на сантиметр. В современных ЗУ предусматриваются средства выявления и исправления ошибок, форматизации, сопряжения и т. п., что раньше выполнялось с помощью внешних ЭВМ.

УДК 681.326

Д. А. Лукьянов

СХЕМОТЕХНИКА УНИВЕРСАЛЬНЫХ ПРОГРАММАТОРОВ ПЗУ

Для работы с ПЗУ в лаборатории необходим универсальный прибор-программатор, позволяющий заносить и считывать информацию как минимум во все типы БИС, перечисленные в таблице *, и блоки на их основе. Такой программатор должен обладать некоторой функциональной избыточностью для работы с новыми ПЗУ и иметь широкие «сервисные» возможности — копирования, редактирования и распечатки содержимого в различных форматах, сравнения и преобразования кодов для удобства трассировки печатных плат, анализа годности БИС. Желательно, чтобы работать с таким прибором мог неспециалист.

Этим требованиям может удовлетворить только программатор со «встроенным интеллектом», а по набору устройств ввода-вывода (магнитные диски, терминалы, печать) сравнимый с микроЭВМ. Здесь нет противоречия — *универсальные программаторы практически всегда становятся периферийным устройством ЭВМ*, основную часть времени используемой для других задач. Эта ЭВМ и реализует вторую компоненту «универсальности» — гибкую, дружественную для пользователя, программную поддержку, предоставляя свои ресурсы программатору.

Несмотря на разнообразие способов записи информации в ПЗУ, все они сводятся к подаче на выводы микросхем последовательности логических сигналов и программирующих импульсов [1]. Цикл записи обычно состоит из выбора адреса ячейки и подготовки БИС к программированию t_1, занесения информации в выбранную ячейку t_2 и восстановления состояния БИС t_3. За циклом записи информации часто следует проверка ее правильности, в зависимости от результатов которой принимается решение о дальнейших действиях.

Итак, универсальный программатор — это специализированный интерфейс ЭВМ, ориентированный на ввод-вывод информации в параллельном коде и генерацию последовательностей импульсов с программируемыми параметрами. Желаемый компромисс между его стоимостью, удобством и универсальностью достигается в том случае, если базовый блок рассчитан на программирование нескольких

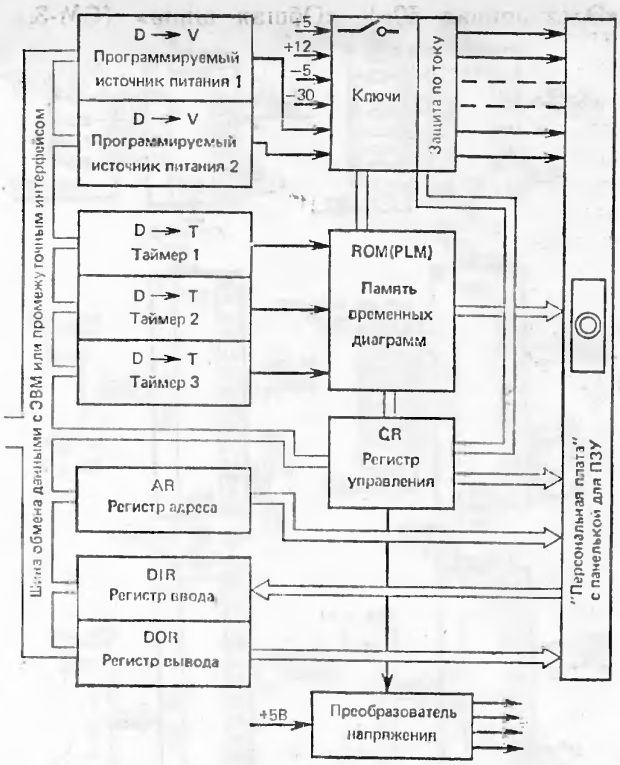


Рис. 1. Структура универсального программатора

наиболее распространенных типов ПЗУ (серии К155, К556, К573), а согласователи уровней и другие узлы для остальных типов размещаются на «персональных» платах (ПП) (рис. 1). Из соображений единообразия контактные устройства для перечисленных типов микросхем располагаются на таких же платах; каждая из плат имеет свой код-идентификатор, который может быть считан ЭВМ. Вместе с программно-управляемым включением питания это позволяет исключить разрушение ПЗУ в результате ошибок оператора.

Существенное значение играет выбор интерфейса для связи с ЭВМ. Промышленные модели программаторов чаще всего имеют стандартные интерфейсы RS232 или IEEE488. Из-за большого числа приемников и передатчиков информации внутри универсального программатора оба интерфейса требуют обработки операций ввода-вывода встроенным микропроцессором. Можно непосредственно передать данные в регистры основных узлов программатора, объединив их внутренним каналом обмена информации и «отобразив» их на пространство ввода-вывода микроЭВМ с помощью промежуточного интерфейса. Так как основные узлы структуры (см. рис. 1) наиболее просто реализуются на БИС серии КР580, то для интерфейса логично принять соглашения канала МП КР580ИК80. Он легко сопрягается с интерфейсами И41 (СМ-1800), МПИ

* См. «Введение» к циклу статей «Учись работать с ПЗУ».

(«Электроника 60»), «Общая шина» (СМ-3, СМ-4) и магистралью крейта КАМАК.

Универсальный программатор можно построить из модулей на основе интегральных микросхем средней и большой степени интеграции. Программируемые однополярные источники питания составляют основу программатора, служат для питания БИС К573 по входу PGM и устанавливают амплитуду прожигающих импульсов для микросхем К155, К556 и К500. Вполне достаточно, если их выходное напряжение будет изменяться в пределах 1...30 В при токе 200...500 мА. Эти параметры легко достижимы с помощью БИС цифро-аналогового преобразователя (ЦАП) КР572ПА1 и параллельного интерфейса КР580ИК55, все порты которого программируются на вывод (управляющее слово 200_8) (рис. 2). Напряжение 1...26 В устанавливается с дискретностью 0,1 В двоичным кодом, записываемым в порты DD1. Точное значение коэффициента пропорциональности выходного напряжения и кода подстраивается с помощью RV1.

Для занесения информации в большинство типов микросхем ПЗУ необходимы импульсы напряжения с крутыми фронтами. Двухтактные электронные ключи VT4, VT6, DD3 должны одинаково хорошо коммутировать любое напряжение ЦАП, поэтому в них применены стабилизаторы тока отпирания VT3, VT5. Аналогично работают ключи записи информации в ПЗУ с плавкими перемычками, в которых дополнительно реализовано ограничение выходного тока на уровне 150 мА. Нижние и верхние плечи ключей соединяются на ПП и управляются различными разрядами DD1, допуская переопределение назначения элементов в зависимости от типа ПЗУ. Для того чтобы после включения питания программатора, когда все линии DD1 находятся в состоянии ввода (это эквивалентно «Лог. 1» на входах DD2), ключи не проводили ток, линия PC0 переводит все выходы DD2 в высокоимпедансное состояние. Одновременно запрещается работа преобразователя напряжения питания программатора.

Напряжения ЦАП на выводы программируемой микросхемы подаются под управлением формирователя временных диаграмм. Если быстродействие ЭВМ и операционная система (ОС) позволяют программно сформировать импульсы длительностью от 0,5...1 мкс, то функции формирователя может выполнять параллельный порт вывода. Однако в ЭВМ с процессорной регенерацией ОЗУ или с мультипрограммными ОС программная реализация временных диаграмм затруднена. Кроме того, одинаковые программные циклы соответствуют различным временам для разных ЭВМ. Поэтому универсальные программаторы часто имеют аппаратный формирователь временных диаграмм. Желательно, чтобы его схемное решение позволяло программно изме-

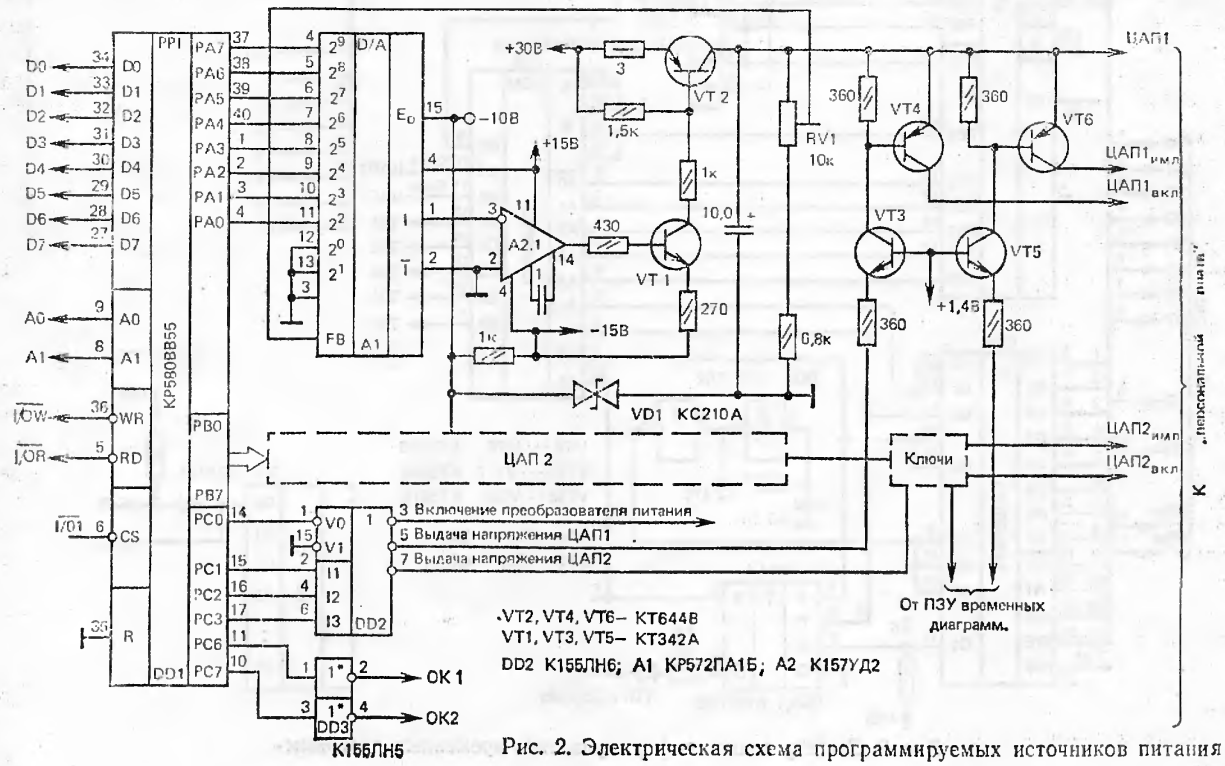


Рис. 2. Электрическая схема программируемых источников питания

нять длительность и взаимное расположение формируемых импульсов.

Хотя формирователь можно реализовать на жестко связанных отдельных логических элементах, коммутирующих импульсы на выход программатора, необходимо более универсальное и регулярное схемное решение, чтобы при появлении новых БИС ПЗУ не нужно было бы изменять схему модуля. Оно будет наиболее простым, если воспользоваться программируемым таймером КР580ВИ53, а необходимые временные диаграммы сигналов (32 варианта) записать в управляющую память (рис. 3), объема которой достаточно для хранения диаграмм. Каждый разряд хранимых в ней чисел соответствует одному ключу или входу ПЗУ. Для выбора необходимого варианта задается адрес страницы ПЗУ, а последовательность адресов ячеек, из которых выбирается текущее состояние сигналов программатора, определяется выходными сигналами трех последовательно включенных таймеров DD2. К сожалению, этот таймер не может генерировать последовательность из трех смежных импульсов (рис. 4), поэтому цикл программирования разбивается не на три, а на пять интервалов, три из которых могут программно устанавливаться в пределах 1...65535 мкс с дискретностью 1 мкс. Триггер DD4.1 вырабатывает признак цикла программирования τ.

Регистры ввода-вывода образуют порты микросхем КР580ИК55, нагрузочной способности которых вполне достаточно для записи

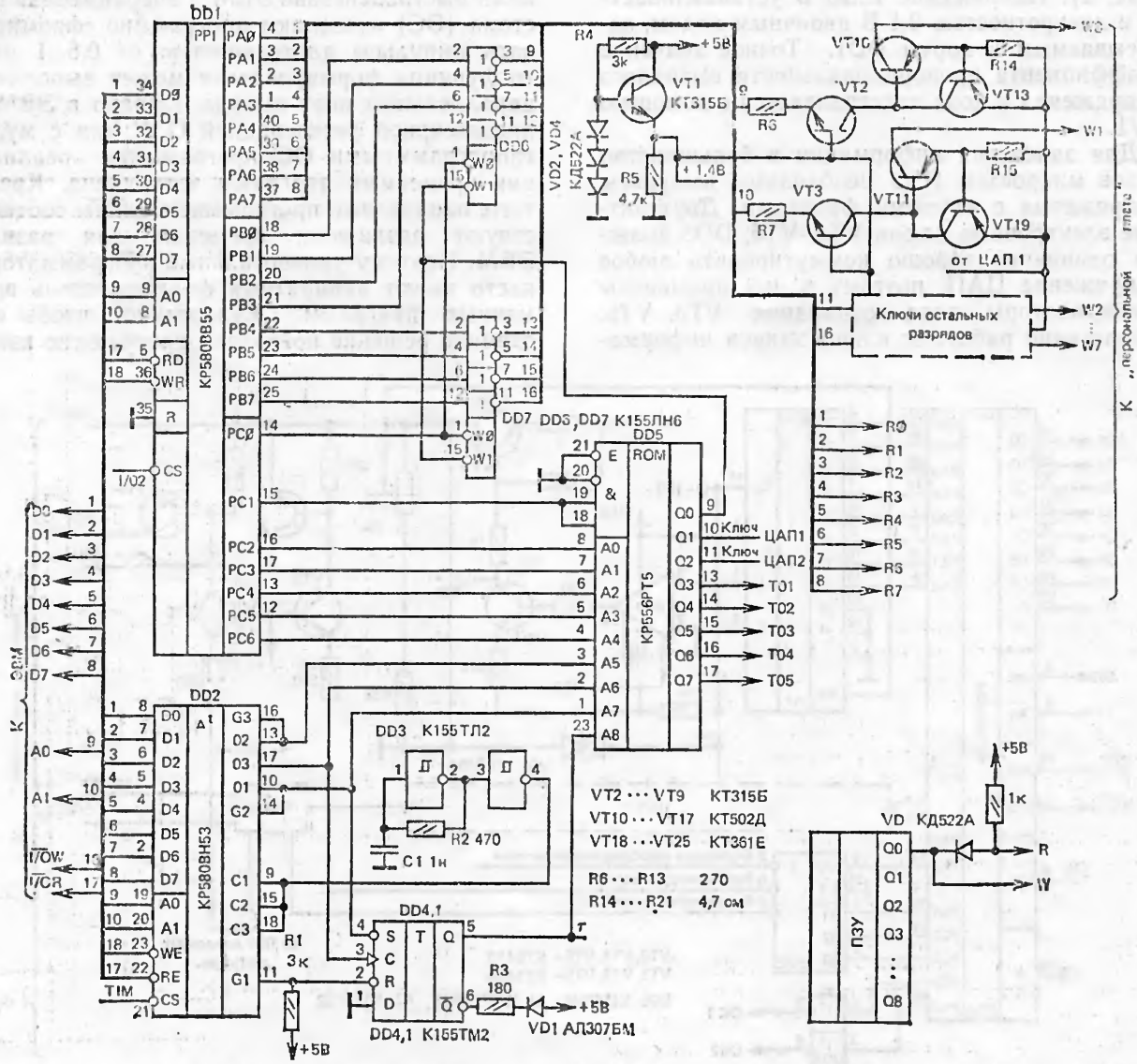


Рис. 3. Регистр записи и формирователь временных диаграмм

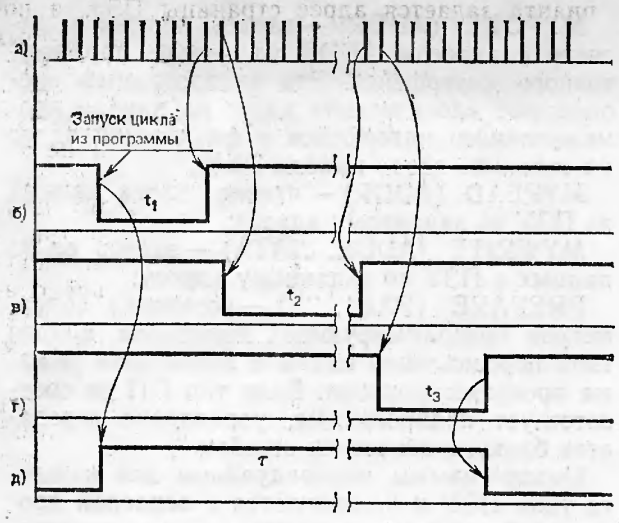


Рис. 4. Временные диаграммы сигналов таймеров: опорная частота *(а)*; выход таймера1 *(б)*; выход таймера2 *(в)*; выход таймера3 *(г)*; выход триггера полного цикла τ *(д)*

в ПЗУ с электрическим и ультрафиолетовым стиранием. Базовый блок осталось снабдить регистром адреса разрядностью не менее 16 и нагрузочной способностью, которая позволяла бы считывать информацию не только из отдельных микросхем ПЗУ, но и из плат ПЗУ. Для этого подойдут буферные регистры К589ИР12 или К155ИР15, легко сопрягаемые с каналом серии КР580.

Даже в минимальном составе базовый блок вместе с персональными платами уже можно применить практически. На ПП для микросхем К573 выводы R0...R7 работают на запись и считывание информации и соединяются с выходами данных ПЗУ. Источником питания входа программирования является ЦАП1, а ТТЛ-входы управления микросхемы соединяются с выходами T01...T05. На ПП для серий К155 и К156 выводы W0...W7 (запись) и R0...R7 (чтение) соединяются вместе (рис. 3,*а*), а выходы $ЦАП1_{вкл}$ и $ЦАП2_{вкл}$ питают вход программирующих импульсов и всю микросхему соответственно. На ПП для микросхем К500РЕ149 находятся преобразователи уровней адреса (ТТЛ/ЭСЛ, К500ПУ124) и данных (ЭСЛ/ТТЛ, К500ПУ125), и ключ входа EN микросхемы. Импульсы прожигания формируются на выходах W0...W3 базового блока. На ПП для микросхем К1601 и К558 имеются ключи напряжений программирования и стирания (—12 В, —26 В).

Дальнейшее расширение возможностей системы и работу с двунаправленной магистралью обеспечат несколько «лишних» портов КР580ИК55, конкретные функции которых определяются ПП и программным обеспечением.

Все узлы объединяются единым каналом обмена информацией, состоящим из восьми линий данных, восьми линий адреса и двух линий управления записью и считыванием. Если адресное пространство программатора является частью адресного пространства ЭВМ, то к регистрам его функциональных узлов можно обращаться как к ячейкам памяти. Для выбора регистра, участвующего в опера-

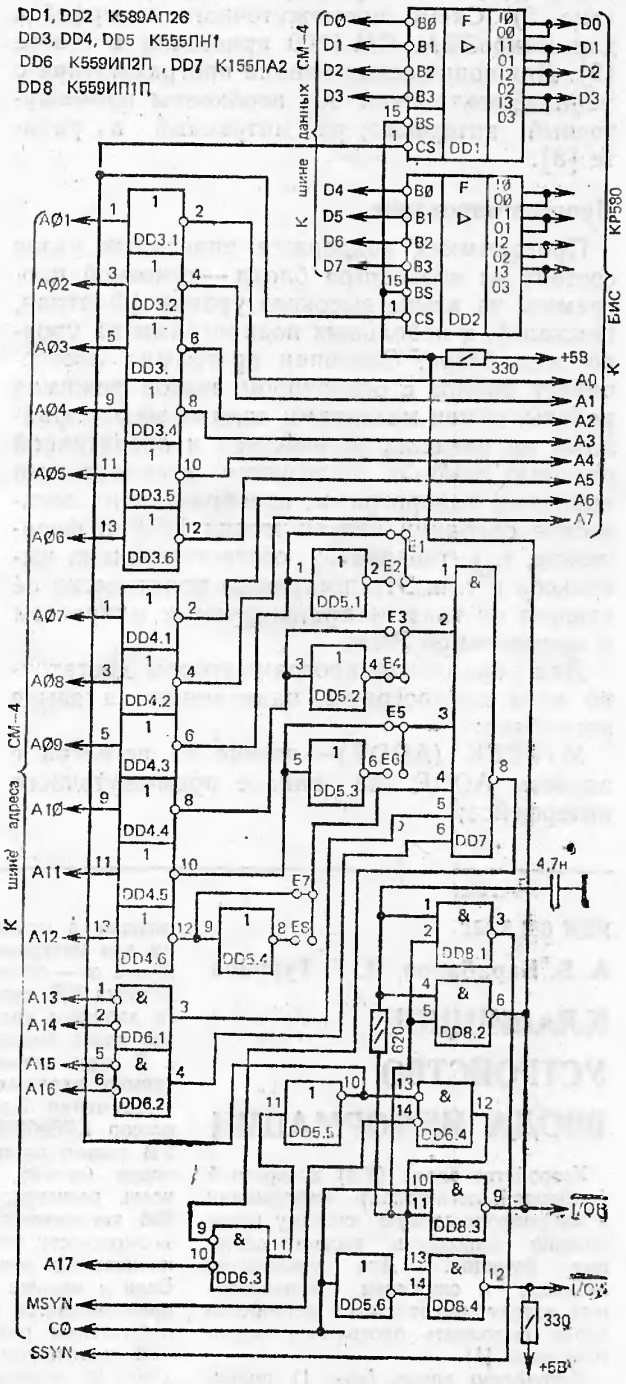


Рис. 5. Схема промежуточного интерфейса на «Общую шину» СМ ЭВМ

ции ввода-вывода, подходящий дешифратор адреса обеспечивает низкий уровень на входе $\overline{CS}$ соответствующей БИС.

Сопряжение канала программатора с «Общей шиной» СМ ЭВМ и мини-ЭВМ «Электроника 100/25» происходит через промежуточный интерфейс, отображающий часть страницы ввода-вывода ЭВМ, заданную перемычками E1...E8, на шину адреса программатора (рис. 5). Схема промежуточного интерфейса для микроЭВМ СМ-1800 приведена в статье [2]. Для сопряжения канала программатора с ЭВМ «Электроника 60» необходим промежуточный интерфейс, рассмотренный в работе [3].

Программирование

Программная поддержка описанных узлов состоит из «базового» блока — основной программы на языке высокого уровня (Фортран, Паскаль), и небольших подпрограмм на уровне ассемблера. Основная программа обеспечивает диалог с оператором, выбор режимов работы, обмен массивами данных между файлами на внешних устройствах и оперативной памятью ЭВМ и пользуется стандартными вызовами подпрограмм, преобразующих логические операции записи-чтения ПЗУ в физические, т. е. генерацию соответствующих импульсов и т. п. Эта программа практически не зависит от типа программируемых микросхем и применяемой ЭВМ.

Для управления программатором достаточно пяти подпрограмм, написанных на языке ассемблер:

MYPEEK (ADDR) — чтение из регистра с адресом ADDR на канале промежуточного интерфейса;

MYPOKE (ADDR) — запись данных в регистр с адресом ADDR на канале промежуточного интерфейса. Эти подпрограммы преобразуют «логический» адрес на канале промежуточного интерфейса в физический адрес на странице ввода-вывода ЭВМ;

MYREAD (ADDR) — чтение слова данных из ПЗУ по заданному адресу;

MYWRITE (ADDR, DATA) — запись слова данных в ПЗУ по заданному адресу;

PREPARE (PAR1, ...) — установка параметров программирующих импульсов, анализ типа персональной платы и подготовка режима программирования. Если тип ПП не соответствует подпрограмме, управление передается блоку диагностики ошибок.

Подпрограммы индивидуальны для каждого типа ПЗУ и компонуются с основной программой на этапе генерации. Такое построение программного обеспечения позволяет уменьшить затраты времени программиста, сохранить единообразие диалога с оператором и облегчить перенос программ на другие типы ЭВМ.

На основе изложенных технических и программных решений построен программатор ПЗУ и его программное обеспечение для работы с микроЭВМ «Электроника 60» и операционной системой РАФОС.

ЛИТЕРАТУРА

1. Дианов А. П., Щелкунов Н. Н. Методика записи информации в ПЗУ.— См. наст. вып. с. 75.

2. Дианов А. П., Щелкунов Н. Н. Модуль записи информации в ПЗУ.— См. наст. вып. с. 80

3. Лукьянов Д. А. КР580 — автоматизация без проблем!— Микропроцессорные средства и системы, 1985, № 1, с. 85.

Статья поступила 15 марта 1985 г.

УДК 681.3.021

А. Б. Барабанов, Н. Г. Турунов

КЛАВИШНОЕ УСТРОЙСТВО ВВОДА ИНФОРМАЦИИ

Устройство ввода (УВ) дискретной (цифровой, логической) информации в микропроцессорную систему часто должно выполнять весьма сложные функции. Для увеличения гибкости и снижения аппаратурных затрат имеет смысл устройство ввода выполнить программно-эмулированным [1].

Устройство ввода (рис. 1) принимает информационные воздействия от оператора и выдает соответствующие сигналы в микропроцессорную систему для интерпретации (выполнения). Пусть a_i — сигнал от контактного датчика КД, тогда в систему выдается двоичное число, определяющее номер этого контактного датчика, т. е. i. В этом случае количество различаемых входных сигналов равно 2^m.

Применяя 8-разрядный микропроцессор КР580ИК80А и присваивая УВ только один 8-разрядный порт ввода ($m=8$), получаем возможность регистрировать сигналы от 256 контактных датчиков. Учитывая возможности оператора, обычно ограничиваются меньшим количеством. Если n меньше или равно m, то УВ представляется регистром, в котором избыточные разряды заземляются.

В клавишном устройстве ввода (рис. 2) количество обслуживаемых КД n больше разрядности порта ввода m, но меньше, чем 2^m. Объединим КД в матрицу размерностью $c \times d$, причем $n \leqslant c \times d \leqslant 2^m$. В это УВ включены также входной регистр с дешифратором и выходной шифратор со своим регистром. Код во входном регистре определяет, какая из групп КД, расположенных вдоль горизонталей матрицы, опрашивается в данный момент. На горизонтали выбранной группы устанавливается сигнал «Лог.0», на остальных горизонталях — «Лог.1». Состояние КД (замкнуто-разомкнуто) на опрашиваемой горизонтали определяет уровни, которые установятся на соответствующих вертикалях и, следовательно, код, который будет зафиксирован в выходном регистре.

Принцип работы выходного шифратора определяет тип применяемых КД: с фиксацией или без фиксации. Если используется обыкновенный (не приоритетный) комбинационный шиф-



Рис. 1. Схема устройства ввода

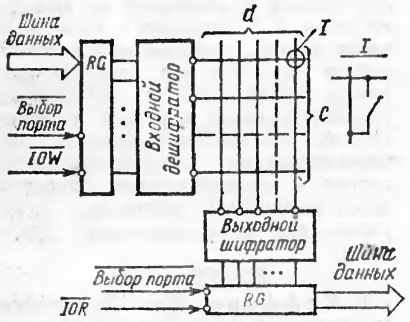


Рис. 2. Схема клавишного устройства ввода

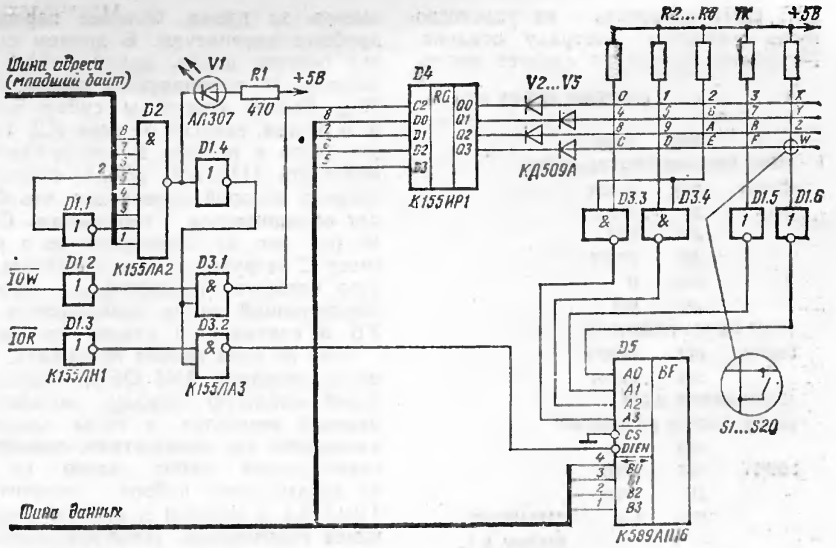


Рис. 3. Принципиальная схема клавишного устройства для ввода шестнадцатеричной информации в микропроцессорную систему на базе БИС КР580ИК80А

ратор, не допускающий нажатия более одного КД, то следует использовать КД без фиксации. Уверенность в том, что замкнут только один из КД на горизонтали, гарантируется более высокой скоростью опроса клавиатуры по сравнению со скоростью реакции человека. Таким образом, нижний предел частоты опроса клавиатуры есть скорость реакции оператора (~10 Гц). Верхний предел частоты опроса определяется продолжительностью переходных процессов в цепях УВ, и чаще всего находится (если это необходимо) опытным путем.

Клавишное УВ (рис. 3) состоит из 20 кнопок и предназначено для ввода шестнадцатеричной информации в микропроцессорную систему на базе КР580ИК80А (первые 16 кнопок) и выполнения дополнительно четырех функций (оставшиеся четыре кнопки).

Входной дешифратор эмулируется программно, и выбранная горизонталь отмечается нулем в старшей тетраде байта, записываемого в порт «FD_{16}» (номер порта выбран произвольно). Дешифратор адреса порта собран на ИС D1.1 и D2. При обращении к порту ввода-вывода, определенному для клавиатуры, готовность устройства показывается светодиодом V1.

Элементы D1.2, D1.3, D1.4, D3.1, D3.2 формируют сигналы переноса кода выбора горизонтали в регистр D4 и считывания отклика клавиатуры с шинного формирователя D5. Диоды V2 ... V5 защищают выходы ИС D4. В байте, считываемом микропроцессором из порта «FD_{16}», младшая тетрада формируется выходным шифратором D3.3, D3.4, D1.5, D1.6. Первые три элемента формируют двоичный код от 001_2 до 100_2 (при замыкании кнопок, предназначенных для ввода шестнадцатеричной информации), а последний показывает состояние управляющей кнопки, которая в данном случае может быть с фиксацией. Устройство шифратора определяется назначением клавиатуры и может быть (независимо от остальной части схемы) изменено. Количество кнопок сокращается простым исключением отдельных узлов матрицы клавиатуры.

В качестве шифратора можно применить ИС приоритетного шифратора (это дает защиту от одновременного нажатия нескольких кнопок в одной горизонтали). Регистры заменяются на любой набор триггеров, а шинные формирователи на любые ИС с третьим состоянием на выходе.

Проектировать дешифратор порта и схему формирования управляющих сигналов следует с учетом сигналов, вырабатываемых применяемой микропроцессорной системой.

Программное обеспечение описанной клавиатуры. Учитывая прозрачность и небольшой объем блок-схемы алгоритма обслуживания клавишного устройства ввода (рис. 4), сразу перейдем к программе (рис. 5) обслуживания клавиатуры в предложенном варианте (шестнадцатеричный ввод и 4 управляющие функции).

Начнем с метки START. Прежде всего необходимо дождаться окончания механического дребезга контактов от предыдущих нажатий и удостовериться в том, что все кнопки отжаты. Для этого надо считать отклик от клавиатуры и командой

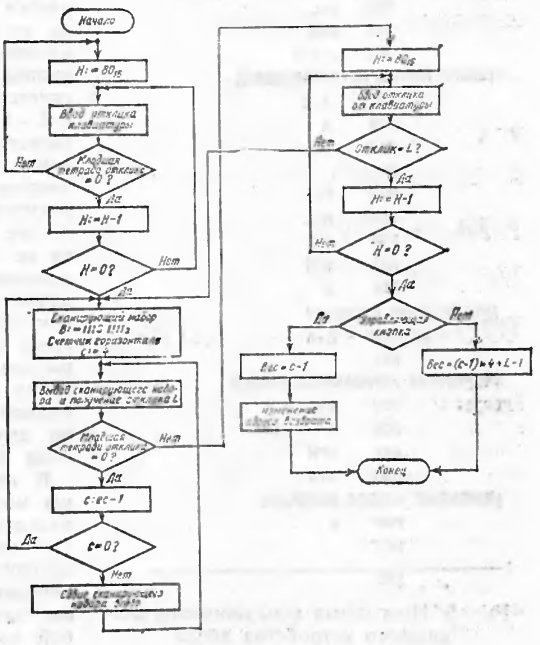


Рис. 4. Блок-схема алгоритма обслуживания клавишного устройства ввода

ANI OFH проверить на равенство нулю младшую тетраду отклика. Подобную процедуру следует проделывать за время, большее периода дребезга клавиатуры. В данном случае счетчик цикла, находящийся в регистре H, инициируется константой 80_{16}. Выбор константы субъективен и, в общем, зависит от типа КД. После этого в регистр B загружается константа 11101111_2 (EF_{16}), старшая тетрада которой определяет, что будет опрашиваться горизонталь C—W (см. рис. 3). Одновременно в регистр C загружается вес горизонтали (это четвертая горизонталь). Затем сканирующий набор записывается в УВ и считывается отклик на него.

```
;                    АССЕМБЛЕР M80/OS 1800
-----------------------------------------------
          ORG     100H
;ЦИКЛ ОЖИДАНИЯ ОТПУСКА КНОПКИ
START:    MUI     H,80H
NO1:      IN      OFDH
          ANI     OFH
          JNZ     START
          DCR     H
          JNZ     NO1
;ЦИКЛ ОПРОСА КЛАВИАТУРЫ
BEGIN:    MUI     B,OEFH
          MUI     C,OЧH
;B-СКАНИРУЩИЙ НАБОР
;C-ВЕС ТЕКУЩЕЙ ГОРИЗОНТАЛИ
          MOU     A,B
LOOP1:    OUT     OFDH
          IN      OFDH
          MOU     L,A     ;СОХРАНЕНИЕ
                          ОТКЛИКА В L
          ANI     OFH     ;ВЫДЕЛЕНИЕ МЛАД
                          ШЕЙ ТЕТРАДЫ
          JNZ     YES1
;ИНАЧЕ ПОЛУЧЕН НУЛЕВОЙ ОТКЛИК
          DCR     C
          JZ      BEGIN
;ИНАЧЕ НЕ ВСЕ ГОРИЗОНТАЛИ ОПРОШЕНЫ
          MOU     A,B
          RLC
          MOU     B,A
          JMP     LOOP1
YES1:     MUI     H,80H
;УСТРАНЕНИЕ ДРЕБЕЗГА
LOOP2:    IN      OFDH
;         CMP     L       ;СРАВНЕНИЕ
                          ; С ОБРАЗЦОМ
          JNZ     BEGIN
;ИНАЧЕ НАЖАТА ВСЕ ТА ЖЕ КНОПКА
          DCR     H
          JNZ     LOOP2
;ПОЛУЧЕННЫЙ ОТКЛИК ЯВЛЯЕТСЯ ВЕРНЫМ
          MOU     A,L
          ANI     08H
          JNZ     CNTR
;ИНАЧЕ КНОПКА НЕ УПРАВЛЯЮЩАЯ
          MOU     A,C
          DCR     A
          RLC
          RLC
          MOU     C,A
          MOU     A,L
          DCR     A
          ANI     OFH
          ADD     C
;ВЕС КЛАВИШИ ВЫЧИСЛЕН
          MOU     B,A
          RET
;ОБРАБОТКА УПРАВЛЯЮЩЕЙ КНОПКИ
CNTR:     MOU     A,L
          DCR     A
          ANI     OFH
          MOU     B,A
;ИЗМЕНЕНИЕ АДРЕСА ВОЗВРАТА
          POP     D
          RET
;-----------------------------------------------
          END
```

Рис. 5. Программа обслуживания клавишного устройства ввода

Если ни одна кнопка не нажата, то после операции ANI OFH, выделяющей младшую тетраду, останется нулевой результат, и тогда следует уменьшить вес горизонтали, сдвинуть сканирующий набор влево (т. е. из предыдущего набора получится 1101110_2) и перейти к опросу следующей горизонтали. Цикл повторяется вплоть до нажатия любой кнопки.

Кнопка нажата — управление передается на команду с меткой YES1, после чего в регистре L находится образец отклика, зафиксировавшего нажатие.

Далее начинается цикл устранения механического дребезга КД. Так как старшая тетрада проверяемого отклика уже записана в регистр D4 (см. рис. 3), то в программе необходимо только считывать байт из порта УВ и сравнивать с образцом. Если один и тот же отклик считан с УВ в течение требуемого промежутка времени, то по нему вычисляется вес нажатой кнопки.

Учитывая функциональное назначение УВ, необходимо проверить, не принадлежит ли кнопка к управляющим, т. е., нет ли единицы в 3-м бите, считая справа. Этот факт проверяется командой ANI 08H. Если после логического умножения остался нулевой результат, то управляющая кнопка не нажата, и вес КД вычисляется по формуле $B=(C-1)\cdot 4+L-1$, где C — вес горизонтали из регистра C, L — младший байт отклика, B — результат, остающийся в регистре B. Умножение на 4 производится двойным сдвигом влево. Если же управляющая кнопка нажата, то ее вес вычисляется по весу горизонтали минус единица, но для отличия от веса информационных кнопок изменяется адрес возврата. Из стека выбирается адрес возврата по шестнадцатеричным кнопкам, и поэтому в конце программы возврат в вызывающую программу происходит по другому, специально для этой цели записанному в стек, адресу.

В качестве вызывающей программы может выступать монитор персональной ЭВМ или абсолютный загрузчик. Если использовать шестнадцатеричные веса кнопок в качестве смещений в таблице переходов, то клавиатура будет полезна для любой микропроцессорной системы с программно-эмулируемым пультом управления. Описанная программа хранит все используемые данные в оперативных регистрах процессора. При изменении команд возврата управления и способа передачи веса управляющей кнопки программа может быть использована в микропроцессорной системе, располагающей ОЗУ ограниченного объема или не имеющей его совсем. Регистр на входе матрицы КД сокращает операции вывода, а шифратор на выходах матрицы КД упрощает вычисление весов кнопок, подобно описанному в работе [2]. Все это сделало программу обслуживания клавишного устройства ввода короткой и эффективной, что особенно выгодно при проектировании микропроцессорных систем с ограниченными аппаратными ресурсами, например, систем управления измерительными приборами.

ЛИТЕРАТУРА

1. Коффрон Дж. Технические средства микропроцессорных систем.— М.: Мир, 1983.—344 с.

2. Хект Н. Упрощение программы обслуживания клавиатуры.— Электроника, 1978, т. 51, № 8, с. 65—66.

Статья поступила 4 января 1985 г.

ОБРАБОТКА ДАННЫХ

— Итак, каковы параметры этого котлована? — осведомился специалист из отдела АСУ.

— Двенадцать на восемь на шесть метров, — отозвался прораб.

— Сейчас подсчитаем. Включаем портативный персональный компьютер — видите, как здорово выглядит, а? Последняя модель! Секундочку, экран разогреется.

— 576 кубов, — произнес прораб.

— Теперь ввожу данные, — объявил специалист. — Двенадцать — девять — стоп. Ошиблись — стираем — итак: двенадцать, восемь, шесть...

— И вот он результат, — обрадованно воскликнул специалист, вытягивая трехметровую ленту из ЭВМ. Обратите внимание — подсчитано точно со скоростью света с учетом мельчайших погрешностей:

576.000.000.000.000 м^3.

— Там всего 576 кубов, — скромно поправил прораб.

— Да что вы говорите! Должно быть — подождите секундочку — действительно, запятая не на месте.

Специалист из отдела АСУ посмотрел восхищенно на прораба, затем удрученно покачал головой и обратился к начальнику отдела кадров:

— Такой, на первый взгляд, целеустремленный, прогрессивный работник, с чувством нового и желанием чему-то научиться, а при профессиональных расчетах пользуется методами каменного века!

Из журнала «Ойленшпигель», 1985, № 18

СПРАВОЧНАЯ ИНФОРМАЦИЯ

Микросхема динамического ОЗУ К565РУ5

Интегральная микросхема К565РУ5 — оперативное запоминающее устройство динамического типа на n-канальных МОП-транзисторах. Корпус 2103.16-5 металлокерамический. Условное графическое обозначение микросхемы приведено на рис 1, назначение выводов показано в табл. 1, структурная схема — на рис. 2,

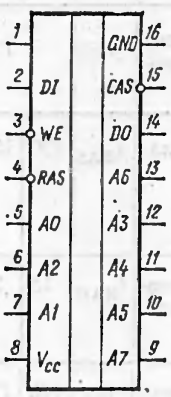


Рис. 1. Условное графическое изображение микросхемы К565РУ5

Таблица 1

Назначение выводов

Вывод	Назначение
1	Свободный
2	Информационный вход D1
3	Вход сигнала «запись-считывание» $\overline{WE}$
4	Вход сигнала выборки строк $\overline{RAS}$
5...7	Адресные входы A0, A2, A1
8	Напряжение источника питания V_{cc}+5В
9...13	Адресные входы A7, A5, A4, A3, A6
14	Информационный выход D0
15	Вход сигнала выборки столбцов $\overline{CAS}$
16	Общий вывод GND

емкость и организация ОЗУ — в табл. 2. Организация ОЗУ в виде матрицы 128×512 обеспечивает полную регенерацию информации за 128 циклов. Для выбора любой из 65536 ячеек памяти требуется подать 16-разрядный код на 8-разрядный адресный регистр. При этом сначала подаются в параллельном коде 8 младших разрядов адреса строки. Код адреса строк (A0...A7) фиксируется сигналом $\overline{RAS}$. 8 старших разрядов кода адреса столбцов A8...A15 фиксируются сигналом $\overline{CAS}$. Когда сигнал $\overline{CAS}$ находится в состоянии «Лог.1», сигнал D0 переходит в состояние высокого импеданса с «плавающим» потенциалом. Кроме того, в режиме записи, когда сигнал $\overline{WE}$ приходит раньше сигнала $\overline{CAS}$, D0 остается в состоянии высокого импеданса в течение всего цикла записи. При считывании D0 переходит в активное состояние (0 или 1) через промежуток времени, равный времени выборки. Во включенном состоянии информационный выход

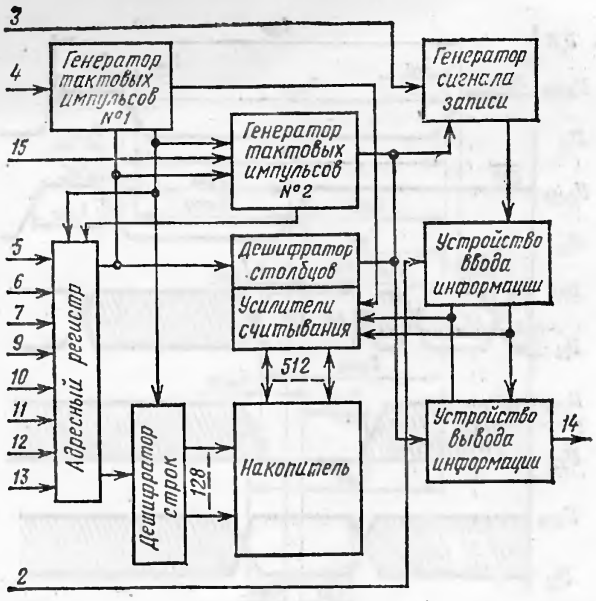


Рис. 2. Электрическая структурная схема

остается до перехода сигнала $\overline{CAS}$ в неактивное высокое логическое состояние. При этом не играет роли, снимается или нет сигнал $\overline{RAS}$.

Таблица 2

Емкость и организация ОЗУ К565РУ5

Микросхема	Емкость, бит	Состояние кода адреса A7	
		при включенном сигнале $\overline{RAS}$	при включенном сигнале $\overline{CAS}$
К565РУ5Б	65536	—	—
К565РУ5В		—	—
К565РУ5Г		—	—
К565РУ5Д		—	—
К565РУ5Д1	32768	0	—
К565РУ5Д2		1	—
К565РУ5Д3	16384	0	0
К565РУ5Д4		1	1

Считываемая информация имеет ту же полярность, что и входная. При подаче сигнала $\overline{CAS}$, по истечении времени t_{RCDmax}, время выборки по отношению к сигналу $\overline{RAS}$ (t_{RAC}) возрастает на величину, t_{RCD}—t_{RCDmax}. Если сигнал $\overline{CAS}$ подается в интервале t_{RCDmin}—t_{RCDmax}, то время выборки t_{CAC} возрастает на величину, равную t_{RCDmin}—t_{RCD}.

Так как в схеме нет выходных буферных регистров для фиксации выходной информации, то в качестве сигнала «выбор микросхемы» могут служить сигналы $\overline{CAS}$ или оба сигнала $\overline{RAS}$ и $\overline{CAS}$. В последнем случае можно реализовать двухкоординатную схему выбора.

Информация считывается без ее разрушения. Регенерация осуществляется обращением к каждой из 128 строк не реже, чем через каждые 2 мс перебором адресов A0...A6. Наиболее удобно выполнять регенерацию в специальном режиме управления одним сигна-

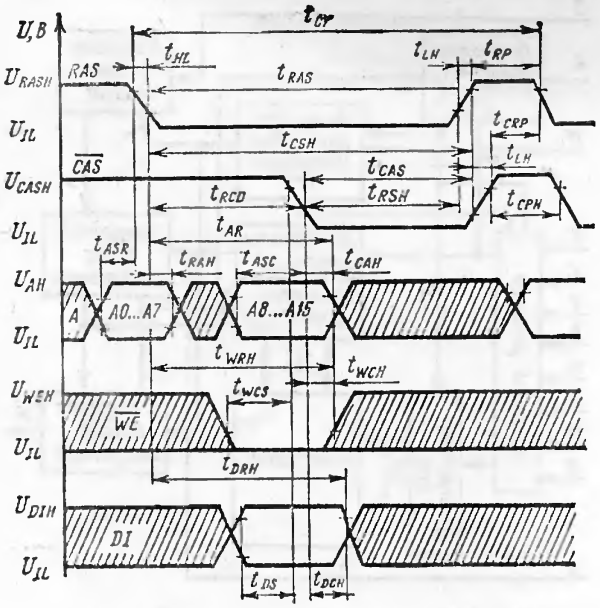


Рис. 3. Режим записи: $t_{HL(LH)}$ для всех сигналов отсчитывается от уровней $U_{IL}=0{,}8$ В до $U_{IH}=2{,}4$ В или от $U_{IH}=2{,}4$ В до $U_{IL}=0{,}8$ В. Заштрихованное поле соответствует сигналу, имеющему произвольное состояние

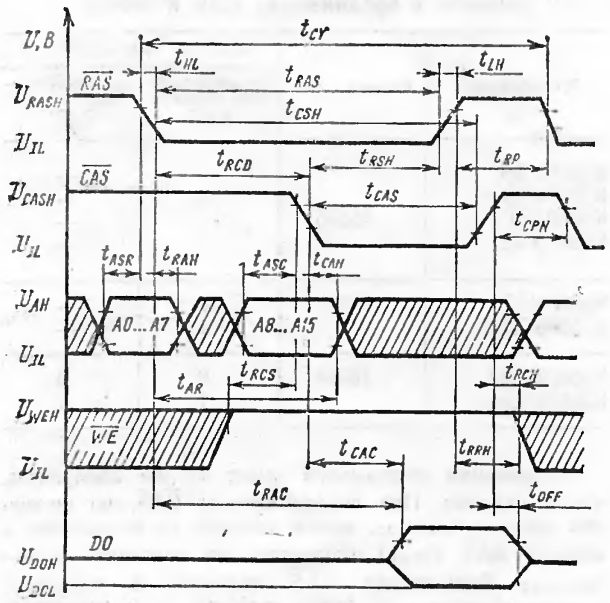


Рис. 4. Режим считывания

лом $\overline{RAS}$, когда сигнал $\overline{CAS}$ находится в неактивном состоянии, так как при этом потребляемая мощность минимальна.

Временные диаграммы работы микросхем приведены на рис. 3...7, а временные параметры сигналов на управляющих входах, обеспечивающие правильную работу,— в табл. 3. Основные электрические параметры представлены в табл. 4, предельные значения допусти-

Таблица 3

Временные параметры сигналов на управляющих входах

Наименование временного параметра	Обозначение	Норма (нс) для микросхемы (не менее) К565РУ5Б	К565РУ5В	К565РУ5Г	К565РУ5Д	Примечание
Время установления сигнала адреса строк относительно RAS	t_{ASR}	⩾0				
Длительность сигнала $\overline{RAS}$	t_{RAS}	120	150	200	250	Не более 10 000 нс (4)
Время удержания сигнала адреса строк относительно $\overline{RAS}$	t_{RAH}	15	20	40	60	(4)
Длительность интервала между сигналами $\overline{RAS}$	t_{RP}	100	120	150	200	
Длительность сигнала $\overline{CAS}$	t_{CAS}	70	90	120	150	Не более 10 000 нс (4)
Время установления сигнала $\overline{RAS}$ относительно $\overline{CAS}$	t_{RCD}	30	35	55	75	(1,4)
Время удержания сигнала $\overline{RAS}$ относительно $\overline{CAS}$	t_{RSH}	70	90	120	150	(4)
Время установления сигнала адреса столбцов относительно $\overline{CAS}$	t_{ASC}	⩾0				(4)
Время удержания сигнала адреса столбцов относительно $\overline{CAS}$	t_{CAH}	25	35	45	60	
Время установления сигнала считывания относительно $\overline{CAS}$	t_{RCS}	⩾0				
Время удержания сигнала считывания относительно $\overline{CAS}$	t_{RCH}	15	20	30	30	
Время установления сигнала DI	t_{DS} t_{DWS}	⩾0				
Время удержания сигнала DI	t_{DCH} t_{DWH}	45	55	80	120	

Наименование временно́го параметра	Обозначение	Норма (нс) для микросхемы (не менее) К565РУ5Б	К565РУ5В	К565РУ5Г	К565РУ5Д	Примечание
Длительность сигнала $\overline{WE}$	t_{WR}	35	45	80	120	
Время удержания сигнала $\overline{WE}$ относительно $\overline{CAS}$	t_{WCH}	45	55	80	120	
Время установления сигнала $\overline{WE}$ относительно $\overline{CAS}$	t_{WCS}	$\geqslant 0$				(2)
Время записи по сигналу $\overline{CAS}$	t_{CWL}	55	70	95	130	(4)
Время записи по сигналу $\overline{RAS}$	t_{RWL}	55	70	95	130	(4)
Длительность фронта	t_{LH} t_{HL}	$\geqslant 3$				не более 35
Длительность интервала между сигналами $\overline{CAS}$ (в страничном режиме)	t_{CP}	70	80	120	160	
Время удержания сигнала CAS относительно $\overline{RAS}$	t_{CSH}	120	150	200	250	
Время удержания сигнала DI относительно $\overline{RAS}$	t_{DRH}	95	115	160	220	
Время удержания сигнала $\overline{WE}$ относительно $\overline{RAS}$	t_{WRH}	95	115	160	220	
Время удержания сигнала считывания относительно $\overline{RAS}$	t_{RRH}	30	40	60	90	
Время установления сигнала $\overline{CAS}$ относительно $\overline{WE}$	t_{CWD}	70	95	120	155	
Время установления сигнала $\overline{RAS}$ относительно $\overline{WE}$	t_{RWD}	120	155	200	255	
Длительность интервала между сигналами $\overline{CAS}$	t_{CPN}	35	40	60	90	
Время удержания сигнала адреса столбцов относительно $\overline{RAS}$	t_{AR}	75	95	125	160	

Примечание: 1—если $t_{RCD} > t_{RCDmax}$, то время выборки относительно сигнала $\overline{RAS}$ (t_{RAS}) будет возрастать на величину, равную $t_{RCD} - t_{RCDmax}$; 2—при $t_{WCS} > t_{WCSmin}$ информационный выход остается в состоянии высокого импеданса; 3 — временные интервалы и фронты управляющих сигналов измеряются относительно уровней входных сигналов: U_{IL}=0,8 В; U_{AH}=2,4 В; U_{WEH}=2,4 В; U_{RASH}=2,4 В; U_{CASH}=2,4 В; 4—параметры обеспечиваются при $t_{(HL)} \leqslant 5$ нс; 5— временные параметры сигналов на управляющих входах микросхем К565РУ5Д1 (Д2, Д3, Д4) соответствуют тем же параметрам для микросхемы К565РУ5Д.

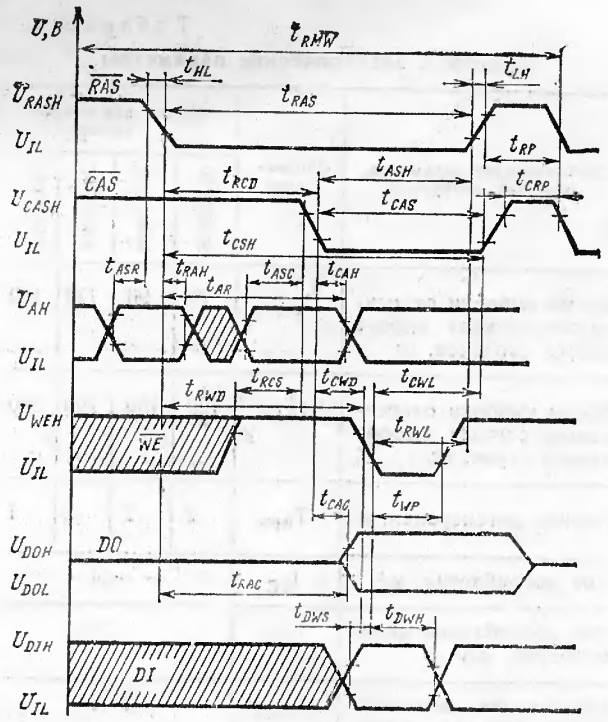


Рис. 5. Режим считывание — модификация — запись (t_{RMW} — время цикла)

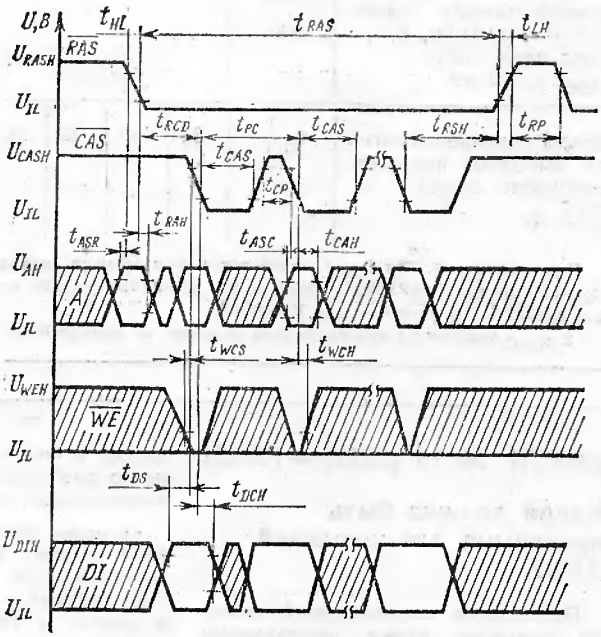


Рис. 6. Страничный режим записи

мых эксплуатационных параметров показаны в табл. 5. Допустимое отклонение напряжения питания от номинального (+5 В) для микросхем К565РУ5Б, К565РУ5В составляет ±10 %, для остальных — ±5 %.

Таблица 4

Основные электрические параметры

Наименование параметра, единица измерения	Обозначение	Норма для микросхемы К565РУ5Б	К565РУ5В	К565РУ5Г	К565РУ5Д
Время выборки относительно сигнала выбора адреса столбцов, нс	t_{CAC}	70	90	120	150
Время выборки относительно сигнала выбора адреса строк, нс	t_{RAC}	120	150	200	250
Период регенерации, мс	T_{RFF}	2	2	2	1
Ток потребления, мА	I_{CC}	$\leqslant 4$			
Ток потребления динамический, мА	I_{CCH}	$\leqslant 45$			
Напряжение высокого уровня сигнала выходной информации, В при токе нагрузки $I_{DOH} \leqslant 2{,}0$ мА	U_{DOH}	$\geqslant 2{,}4$			
Напряжение низкого уровня сигнала выходной информации, В при токе нагрузки $I_{DOL} \leqslant 4{,}0$ мА	U_{DOL}	$\leqslant 0{,}4$			
Время сохранения сигнала выходной информации после сигнала $\overline{CAS}$, нс	t_{OFF}	35	40	60	90

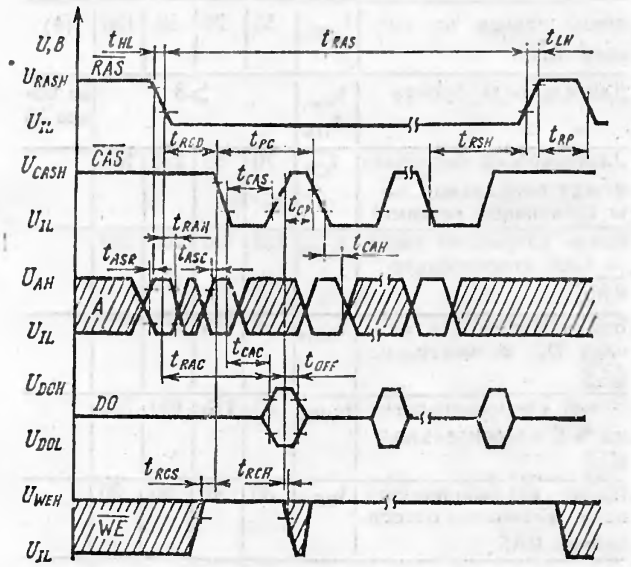
Примечание. 1. Ток I_{CCH} измеряется при времени цикла $t_{CY} \geqslant 230$ мс для К565РУ5Б; $t_{CY} \geqslant 280$ мс для К565РУ5В; $t_{CY} \geqslant 360$ мс для К565РУ5Г; $t_{CY} \geqslant 460$ мс для К565РУ5Д.

2. t_{CAC} измеряется при емкостной нагрузке на выходе 50 пФ.

Таблица 5

Основные эксплуатационные параметры

Наименование параметра, единица измерения	Обозначение	Норма не менее	Норма не более
Напряжения питания относительно общего вывода U_{CND}, В	U_{CC}	0	6,5
Напряжение на выводах микросхемы A0...A7, CAS, RAS, WE, DI относительно общего вывода U_{GND}, В	U	—1,0	6,5
Длительность фронта, нс	t_{LH}; HL	—	50
Емкость нагрузки, пФ	$C_н$	—	100



$$t_{PC} = t_{CAS} + t_{CP} + t_{CASHL} + t_{CASLH}$$

Рис. 7. Страничный режим считывания

БИНТИ № 19 (2212) — (2213)

Какой должна быть программа для школьной ЭВМ?

Применение компьютерной техники открывает перед школьниками большие возможности, которые пока используются далеко не полностью, так как специалисты, составляющие учебные программы для школьных ЭВМ, как правило, не знакомы со спецификой учебного процесса, а учителя недостаточно хорошо разбираются в электронно-вычислительной технике.

Авторам учебных программ для ЭВМ следует познакомиться с психологическими особенностями детей, теорией и практикой обучения в школе, а учителям, в свою очередь, надо изучать электронно-вычислительную технику.

Составляя учебную программу для ЭВМ, надо учитывать:

— насколько эффективной будет данная программа;

— потребуется ли для ее выполнения не только запоминание, но и осмысление пройденного материала;

— насколько нова и оригинальна концепция программы;

— не утратит ли программа свою ценность и не потеряет ли школьник к ней интерес при повторном использовании;

— насколько интересна программа;

— не может ли она испугать или задеть чувство собственного достоинства школьников.

РЕФЕРАТЫ СТАТЕЙ

УДК 681.3

Лопатин В. С., Борисенков В. Д., Юрочкин А. Г., Скокленев А. Г., Баранов Н. Д. **Диспетчер памяти микроЭВМ «Электроника МС 1213».**— Микропроцессорные средства и системы, 1985, № 3, с. 3.

В краткой форме рассмотрены структура и особенности работы диспетчера памяти микроЭВМ «Электроника МС 1213», совместимой с мини-ЭВМ СМ ЭВМ и «Электроника 100/25». Диспетчер позволяет организовать мультипрограммный режим работы и адресоваться к памяти объемом до 256К байт.

УДК 681.322.1

Барышников В. Н., Воронов М. А., Гиглавый 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.

Даны аппаратные и программные характеристики персональной ЭВМ «Ириша», состоящей из системного блока, клавиатуры и доработанного цветного или черно-белого телевизора. В таком комплекте она монтируется на рабочем месте учащегося школы, ПТУ или техникума и связана с машиной преподавателя с помощью локальной сети. ПЭВМ преподавателя оснащена дополнительно накопителем на гибких магнитных дисках и печатающим устройством. Рабочие языки программирования: Бейсик, Паскаль, Фортран, СИ и др.

УДК 681.322.181.5+681.322.066+681.3001.18

Громов Г. Р. **Профессиональные приложения персональных ЭВМ.**— Микропроцессорные средства и системы, 1985, № 3, с. 9.

Рассматриваются структура и динамика развития сектора ПЭВМ мировой индустрии обработки данных, основные области профессиональных приложений ПЭВМ: автоматизация учреждений, информационно-измерительные системы, САПР, АСУ ТП и др. Обсуждаются технические характеристики и архитектурные особенности первых трех поколений ПЭВМ. Выделен класс двухпроцессорных ПЭВМ типа «кентавр», используемых на ВЦКП для оптимизации процесса разработки и автономной отладки программ. Формулируются требования к надежности и уровню сервиса ПЭВМ.

УДК 681.3.06

Кочетков Г. Б. **Компьютерные игры: свет и тени.**— Микропроцессорные средства и системы, 1985, № 3, с. 16.

Компьютерные игры являются важным средством расширения масштабов внедрения вычислительной техники. Описывается история игр и их роль в процессе развития информационной технологии.

УДК 681.322.068

Корчак А. Е. **Интерпретатор языка программирования Бейсик/F для микроЭВМ.**— Микропроцессорные средства и системы, 1985, № 3, с. 21.

Описывается язык программирования Бейсик/F, предназначенный для решения широкого круга задач на микроЭВМ. Язык Бейсик/F имеет средства модульного и структурного программирования, развитую файловую систему. Интерпретатор языка Бейсик/F реализован в операционных системах СР/М, МикроДОС, ОС-1800, ISIS-II, ДОС-1800, DOS/F.

УДК 681.3.06—519.68

Варсанофьев 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.

Подробно описывается внешний вид оригинальной высокоинтерактивной системы для ввода, редактирования и выполнения программ, задач и упражнений из учебника «Основы информатики и вычислительной техники» для 9 класса средней школы

УДК 621.317:681.3.62; 621.3.049

Сумин В. В., Кислицын А. Б., Воробьев В. И., Савин А. Г. **Простейшие программируемые измерительные приборы на базе микроЭВМ.**— Микропроцессорные средства и системы, 1985, № 3, с. 33.

Рассматриваются структура, организация и программное обеспечение базового устройства цифрового отсчета и обработки на однокристальной микроЭВМ «Электроника С5-31» (микросхема К586ВЕ1). Устройство можно настраивать на разрабатываемый вариант прибора путем установки сменного ПЗУ с требуемой программой и сменной схемы сопряжения с датчиком.

УДК 681.326

Тарасов Н. А. **Устройство обработки информации с малым энергопотреблением.**— Микропроцессорные средства и системы, 1985, № 3, с. 46.

Описано устройство обработки информации с малой потребляемой мощностью, которое может работать в жестких климатических условиях и предназначено для встраивания в малогабаритную радиоэлектронную аппаратуру. Приведены принципиальные электрические схемы и технические характеристики разработанного устройства.

УДК 681.327.8.06

Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г. **Принципы организации и работы дисплеев на основе БИС КР580ВГ75.**— Микропроцессорные средства и системы, 1985, № 3, с. 51.

Исследуются принципы организации, режимы работы, особенности программирования и взаимодействие отдельных функциональных блоков дисплея, построенного на основе трех программируемых БИС: контроллера электронно-лучевой трубки КР580ВГ75, контроллера прямого доступа к памяти КР580ВТ57 и таймера КР580ВИ53.

УДК 681.327.13

Зеленко Г. В. **Дисплей для бытовой персональной ЭВМ.** Микропроцессорные средства и системы, 1985, № 3, с. 60.

Рассмотрены функциональное назначение и особенности применения контроллера ЭЛТ КР580ВГ75. Приведена схема формирования видеосигнала для телевизора, используемого в качестве дисплея бытовой персональной ЭВМ.

УДК 681.32

Лукьянов Д. А. **Схемотехника универсальных программаторов ПЗУ.**— Микропроцессорные средства и системы, 1985, № 3, с. 84.

Рассматриваются способы построения устройств для занесения информации в ПЗУ (программаторов) на современной элементной базе. Расширяющаяся номенклатура БИС ПЗУ требует их максимальной универсальности, которая наиболее полно реализуется в тех случаях, когда программатор становится периферийным устройством мини- или микроЭВМ, а его параметры могут перестраиваться с помощью управляющей программы.

UDC 681.3.06—519.68

Varsanofjev D. V., Kushnirenko H. Л., Lebedev G. V. **E-Practicum — the Software for Educational Course on Informatics and Computers.**— Microprocessor Devices and Systems, 1985, N 3, p. 27.

The paper considers in details the original system for dealing with programs, problems and exercises of the textbook "Informatics and Computer Science Principles" (the 9-th grade). The system has been used for training teachers of Moscow schools at the Moscow State University in summer 1985.

UDC 621.317:681.3.62;621.3.049

Sumin V. V., Kislitsin A. B., Vorobjev V. I., Savin A. G. **Elementary Programmable Measuring Instruments on the Basis of Microcomputer.**— Microprocessor Devices and Systems, 1985, N 3, p. 33.

The paper describes the structure, organization and software for the digital measurement system based on monocrystal microcomputer "Electronica C5-31" (IC — K586BE1). Tle device may be adapted to particular instrument by installation of changeable ROM and connection scheme.

UDC 681.326

Tarasov N. A. **Microprocessor Device with Small Energy Consumption.**— Microprocessor Devices and Systems, 1985, N 3, p. 46.

The author describes singleboard 8-bit microcomputer with small energy consumption. It can work in extreme climatic conditions and is intended for equipment of small sized radio apparatus. Principal electrical diagrams and technical features are described.

UDC 681.327.8.06

Panfilov D. I., Romanenko O. A., Safanjuk V. S., Sharonin S. G. **The Principles of Display Control on the Basis of VLSI KR580VG75.**— Microprocessor Devices and Systems, 1985, N 3, p. 51.

The paper describes organization, working conditions, peculiarities of programming and interaction of separate functional blocks of display constructed on the basis of three programmable VLSI-s: CRT controller KR580VG75, DMA controller KR580VT57 and timer KR580VI53.

UDC 681.327.8.06

Zelenko G. V. **Display for a Home Personal Computer on the Basis of KR580VG75 Controller.**— Microprocessor Devices and System, 1985, N 3, p. 60.

Functional features and application of CRT controller KR580VG75 are considered. Videosignal formation diagram is presented for driving TV connected to a home personal computer.

UDC 681.32

Lukyanov Dm. A. **Multipurpose PROM programmer design.**— Microprocessor Devices and Systems, 1985, N 3, p. 84.

Circuit design for PROM programmers based on microprocessor electronic components is discussed. Maximum flexibility of PROM programmer is necessary for creative work with still increasing variety of PROM LSI's.

Basic hardware of "PRO-MULTIPLEX" PROM programmer suitable for K155, K556, K573, K1601, Intel 27XX and most other types is given for illustration.

UDC 681.3

Lopatin V. S., Borisenkov V. D., Yurochkin A. G., Skoklenev A. G., Baranov N. D. **Memory Management System for "Electronica MC 1213".**— Microprocessor Devices and Systems, 1985, N 3, p. 3.

The paper considers in a brief form the structure and technical features of memory management system for "Electronica MC 1213" which is compatable with minicomputers SM and "Electronica 100/25". Memory management system makes it possible to organize multiprogramming and addressing of upto 256 Kbyte memory.

UDC 681.322.1

Baryshnikov V. N., Voronov M. A., Giglavy A. V., Panachev F. I., Romanov V. Yu., Titov O. F. "IRISHA" — Personal Computer for Informatics and Computer Classrooms.— Microprocessor Devices and Systems, 1985, N 3, p. 5.

The paper presents the description of hardware and software features of the personal computer "IRISHA". The computer consists of the system unit, a keyboard and a modified colour or black-and-white TV. It is intended for installation in a classroom of an ordinary school or technical college. Connection will be provided by means of a local network. The latter is equipped by floppy-disk and printer. The working languages are Basic, Pascal, Fortran, C etc.

UDC 681.322.181.5+681.322.066+681.31.18

Gromov G. R. **Professional Application of Personal Computers.**— Microprocessor Devices and Systems, 1985, N 3, p. 9.

The paper considers the structure and dynamics of personal computers sector of the data processing industry, the main areas of their professional application including office automation, measuring systems, CAD/CAM et al. The author discusses technical features and architecture of the first 3 generations of personal computers. Special attention is given to the dual-processor system "centaur", which is used for optimizaton of program development and debugging. The problems of PC service and reliability are discussed.

UDC 681.3.06

Kochetkov G. B. **Computer Games: Light and Shades.**— Microprocessor Devices and Systems, 1985, N 3, p. 16.

Computer games are considered as important means in extending application of computers. The author describes the history of computer games and their role in information technology development.

UDC 681.322.068

Korchak A. E. **Programming language BASIC/F interpreter for microcomputers.**— Microprocessor Devices and systems, 1985, N 3, p. 21.

The article presents new programming language BASIC/F for the wide-range microcomputer applications. BASIC/F language contains modular and structured programming tools, extended file system. BASIC/F interpreters designed for such operating systems as CP/M, MicroDOC OS-1800, ISIS-11, DOS-1800, DOS/F are considered.

МИКРОПРОЦЕССОРНАЯ ТЕМАТИКА В КНИГАХ ИЗДАТЕЛЬСТВА „МИР"

Ряд книг издательства «Мир» имеет целью ознакомить советских читателей с зарубежным опытом разработки и применения вычислительной техники, в том числе микропроцессорных средств.

Для технологов электронной промышленности издательство выпустило в свет книги И. Броудая, Дж. Мерея «Физические основы микротехнологии» (1985) и У. Тилла, Дж. Лаксона «Интегральные схемы: материалы, приборы изготовление» (1985). В 1986 г. будут изданы монография С. Мьюрарка «Силициды для СБИС», двухтомник «Технология СБИС» и посвященный новейшему направлению электроники сборник «Молекулярные электронные устройства», в котором рассматриваются вопросы прохождения сигналов в молекулярных агрегатах и их переключения, материалы для изготовления молекулярных устройств и соответствующая микротехнология.

Разработчикам систем обработки данных на базе микропроцессорных средств адресованы книги С. Мурога «Системное программирование сверхбольших интегральных схем» (1985) и Э. Клингмана «Проектирование специализированных микропроцессорных систем» (1985). Г. Майерс в двухтомнике «Архитектура современных ЭВМ» (1985) рассматривает принципы построения современных ЭВМ, ориентированных на конкретные языки и многопроцессорные системы с динамически перестраиваемой архитектурой.

Трехтомное справочное руководство «Компьютеры» охватывает основные проблемы построения компьютеров, периферийного оборудования и соответствующего программного обеспечения (ПО). Дано описание и сравнение диалоговых языков (АПЛ, Бейсик) и алгоритмических (Кобол, Фортран, ПЛ/1, Паскаль), а также принципов организации ПО на базе ассемблера. Приведено полное программное обеспечение микропроцессорных систем Интел 8080 и 8085. Даны примеры применения микроЭВМ в средствах машинной графики, устройствах для распознавания речи и робототехнических системах.

В 1986 г. появится книга М. Фридмена и Л. Ивенса «Проектирование систем с микрокомпьютерами», посвященная методике проектирования микрокомпьютера как единому процессу создания программных и аппаратных средств. При составлении программ на языках высокого уровня рекомендуется модульный принцип, упрощающий процесс их отладки. Даны описания вычислительных систем, применяемых за рубежом в АСУ ТП.

Р. Трейстер в монографии «Персональный компьютер фирмы ИБМ» характеризует устройство этого наиболее популярного за рубежом компьютера, особенности программирования и эксплуатации, приводит технические характеристики и примеры использования IBM PC.

В книге Л. Пула «Работа на персональном компьютере» (1986) рассмотрена методика работы на персональных компьютерах, снабженных необходимым числом периферийных устройств; приведены конкретные примеры программирования на языке Бейсик; описаны способы ввода цифровой и графической информации с помощью дисплея, а также вывода информации на печатающие и громкоговорящие устройства. Много внимания уделено запуску и редактированию программ.

Программистам будет интересна работа К. Зиглера «Методы системного программирования» (1985). Применительно к решению сложных задач автор рассматривает системное программирование на языках Ада, Алгол, Фортран, ПЛ/1, CPSS и др. В монографии Р. Фаулджера «Программирование встроенных микропроцессоров» (1985) анализируются методы программирования на языках Ада, Бейсик, Паскаль и др. применительно к встроенным микропроцессорам, работающим в реальном масштабе времени. В центре внимания — количественные показатели оценки качества программного обеспечения.

Тех, кто **применяет микропроцессорные средства** в робототехнике, конечно, заинтересуют книги П. Андре и др. «Конструирование роботов» (1986), в которой даны принципиальные схемы управления, и Ф. Куафе «Взаимодействие робота с внешней средой». Ф. Куафе анализирует схемы построения датчиков и их групп для решения различных технологических задач, средства очувствления роботов и переработки информации, задачи идентификации объектов и способы выделения их контуров, позволяющие создавать адаптивные робототехнические системы с элементами искусственного интеллекта.

В монографии Р. Эйреса и С. Миллера «Перспективы развития робототехники» (1986) рассмотрены возможности применения существующих и разрабатываемых в США и Японии робототехнических систем для комплексной автоматизации производства, добычи полезных ископаемых, подъема затонувших судов, выполнения сборочно-сварочных работ в космическом пространстве, а также в исследовании планет солнечной системы. Обсуждаются предельные возможности роботов с программным управлением, адаптацией и элементами искусственного интеллекта.



1р.10к. Сусликов Индекс 70588

ВНИМАНИЕ! ИДЕТ ПОДПИСКА НА 1986 ГОД

Подписка на журнал «Микропроцессорные средства и системы» принимается отделениями «Союзпечать» без каких-либо ограничений на всей территории Советского Союза, а за рубежом в отделениях «Межкнига».

Журнал позволит Вам первым узнавать о новинках в области микропроцессорной техники, а также регулярно знакомиться со схемами и примерами программ для самостоятельного создания конкретных устройств автоматизации на базе микроЭВМ и микропроцессоров.

Как показал анализ ответов на «анкету читателя», журнал стал настольным, рабочим пособием для профессионалов в области ЭВМ и программирования, а также представляет непосредственный практический интерес для самого широкого круга специалистов отраслей народного хозяйства, делающих первые шаги в новый для них мир микропроцессорных средств автоматизации.

Циклы статей под рубриками «Учись работать с ПЗУ», «Ваша первая микроЭВМ» и другими из раздела «Учебный центр» имеют целью помочь нашим читателям на простых примерах практически освоить поистине неисчерпаемые возможности микропроцессорной техники.

На страницах журнала регулярно обсуждаются актуальные проблемы программирования, новейшие тенденции развития информационной технологии. Раздел «Трибуна ученого», дискуссии ученых и специалистов по наиболее острым проблемам развития вычислительной техники помогают читателям контролировать «горячие точки» науки и технологии в этой бурно развивающейся области.

Напоминаем, что в розничную продажу журнал не поступает, а достать его в библиотеке, как утверждают читатели, по разным причинам не успевшие в этом году оформить подписку, оказывается весьма не просто...

Журнал «Микропроцессорные средства и системы» — Ваш первый путеводитель в мир микропроцессорной техники — мир техники будущего! Не забудьте своевременно оформить подписку на 1986 год!

Наш индекс по каталогу «Союзпечать» — 70588. Цена годового комплекта журналов — 4 руб. 40 коп.