



МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

4 | 1987

ISSN 0233-4844

Электроника МК 85: полная QWERTY-клавиатура (язык программирования — Бейсик) в габаритах карманного микрокалькулятора

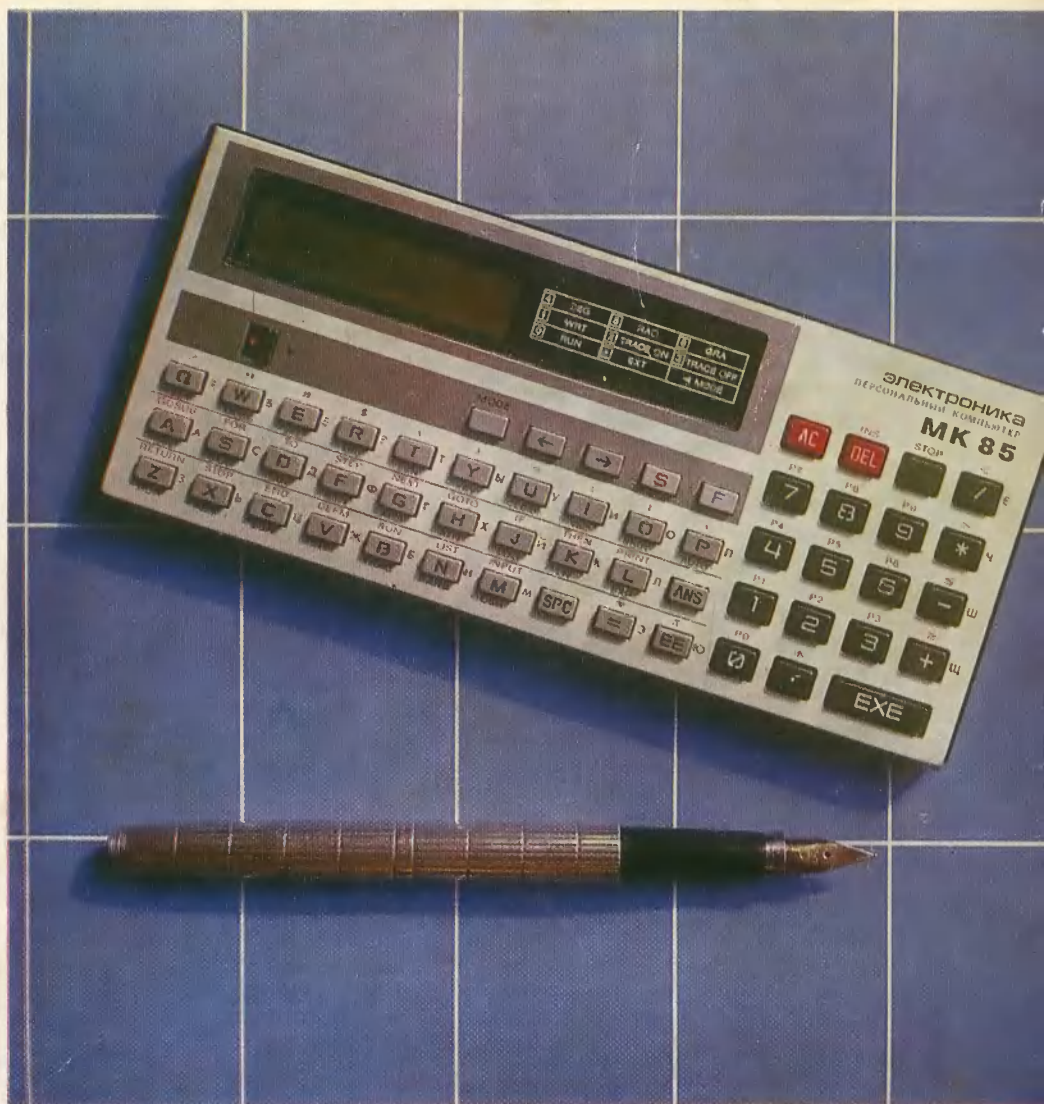
Микропроцессорное матричное печатающее устройство «Электроника МС 6307» для микроЭВМ с интерфейсами «Стык С2» и «20 мА токовая петля»

Машинная графика: стирается грань между алфавитными и цифровыми дисплеями

Инженер оформляет заявку на изобретение: система безбумажной патентной технологии

Инструментальные средства проектирования прикладных систем без привлечения профессиональных программистов

Комплект БИС серии КР1828 для тестового диагностирования микропроцессоров, устройств памяти, схем произвольной логики



МАШИННАЯ ГРАФИКА

Ю. М. Баяковский, В. А. Галактионов

Машинная графика — это раздел информатики, в котором изучаются методы и средства синтеза, хранения и вывода изображений с помощью ЭВМ.

Рис. 1 иллюстрирует использование компьютеров и машинной графики в кинематографии: показано изображение космолана, сгенерированное компьютером для съемок фантастического фильма. Программная модель космолана состоит из 400 тыс. многоугольников. Для обработки изображений такой сложности требуются огромные вычислительные мощности. Поэтому некоторые кинофирмы применяют для этих целей новейшие суперкомпьютеры, в том числе CRAY X-MP с быстродействием 200 млн. операций в секунду. Однако даже при использовании подобных ЭВМ синтез каждого кадра требует свыше 5 минут счетного времени. Таким образом, на создание 12-минутного фильма потребуется свыше месяца работы суперкомпьютера.

Создание программного обеспечения для производства компьютерных фильмов и эксплуатация супер-ЭВМ стоят, разумеется, недешево. Однако, как сообщается, снимаются такие фильмы вдвое быстрее, стоят в 2—3 раза дешевле и требуют в 4—6 раз меньших затрат человеческого труда по сравнению с традиционными методами, предусматривающими построение миниатюрных моделей и последующую обработку снятого киноматериала. Кроме того, методы компьютерного моделирования и машинной графики позволяют создавать образы, существенно более сложные, чем те, с которыми обычно работают кинематографисты.

Создание реалистических изображений



Рис. 1

Одним из методов построения высокореалистичных изображений сложных пространственных сцен является метод трассирования лучей. Его основная идея состоит в отслеживании индивидуальных световых лучей для определения, какие из этих лучей попадают в точку зрения. Однако реально из источника света исходит бесконечное количество лучей, и только малая их часть достигает точки зрения. Поэтому в алгоритме световые лучи отслеживаются как бы в обратном направлении: исходят из точки зрения и проходят через каждый пиксел на картинной плоскости (т. е. плоскости экрана) до пересечения с поверхностью. Если поверхность непрозрачная, то в зависимости от физических свойств материала, из которого она изготовлена, часть луча отражается, а оставшаяся часть поглощается веществом. Если отраженный луч будет

направлен на источник света, то образуется блик, в противном случае в качестве источника освещения будет выступать диффузный свет. Если же луч попадает на другой объект, то процесс повторяется.

Наиболее интересна ситуация, когда луч попадает на прозрачный объект. В этом случае часть луча отражается (зеркально и диффузно), часть — преломляется и проходит через объект, а оставшаяся часть поглощается веществом. Траектории двух образовавшихся лучей отслеживаются аналогичным образом. Цвет соответствующего пиксела будет определяться комбинацией интенсивностей этих двух лучей с учетом диффузной составляющей отраженного света. Далее лучи могут встречать на своем пути другие прозрачные и непрозрачные поверхности, вновь разделяться и т. д.

(Продолжение см. на 3 с. обл.)

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года

ММП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 4 / 1987 МОСКВА

**СОДЕРЖАНИЕ
МИКРОПРОЦЕССОРНАЯ
ТЕХНИКА**

**ПЕРСОНАЛЬНЫЕ
КОМПЬЮТЕРЫ**

**ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ**

**ИНЖЕНЕР ОФОРМЛЯЕТ
ЗАЯВКУ НА
ИЗОБРЕТЕНИЕ**

МАШИННАЯ ГРАФИКА

УЧЕБНЫЙ ЦЕНТР

МИКРОКОНТРОЛЛЕРЫ

**СПРАВОЧНАЯ
ИНФОРМАЦИЯ**

Ершов А. П. — Колонка редактора	2
Жуков Н. Н., Коризенков А. С., Офицеров В. Н. — Печатающее устройство «Электроника МС 6307»	3
Габдуллин Р. Р., Горемыкин В. В., Горяшко А. П., Косов Л. С., Луквич Г. А., Миронов В. Г. — Комплект БИС для встроенного диагностирования микропроцессорных систем	4
Лемко Л. М., Гладков В. В., Ермаков С. В., Жуков В. Н. — Персональный микрокомпьютер «Электроника МК 85»	10
Лопатин В. И., Старовойтов Ю. Н. — Программирование персонального компьютера «Электроника МК 85» с использованием языка Бейсик	13
Казменко С. В. — Автоматизация проектирования систем на основе объединения концепций макетирования и «программирования без программиста»	16
Григорьев А. Г. — Диалоговый дисассемблер для загрузочных модулей операционной системы РАФОС	19
Григорьев А. Г. — Адаптированная операционная система АДОС для СМ ЭВМ	23
Долманов В. Г., Каюфьев А. В. — Структурные особенности интерпретирующей системы Бейсик/РАФОС	25
Селицкий С. С., Сыркин М. Ю. — Процедура перемещения частей загрузочного модуля для микропроцессора КР580ИК80	28
Воржев А. В., Зверков Б. С., Кикоть А. И., Яковлева Е. В. — Интегрированная система разработки кросс-систем на базе ДВК-2М для микропроцессоров	30
Сарьян В. К., Смолич Г. Г. — Взаимодействие больших и малых баз данных в автоматизированной системе оформления заявок	37
Сарьян В. К., Корытов В. В. — Увеличение времени жизни диалоговой системы по оформлению первичных документов по заявкам	40
Блинников В. И., Минашин В. П., Бедин Б. Д., Сарьян В. К. — Система безбумажной технологии делопроизводства по заявкам на изобретения	42
Демин А. П., Харитонов Г. И. — Устройство отображения графической информации на алфавитно-цифровом дисплее	43
Овчинников Л. Г., Сороченко Н. И. — Устройство индикации	46
Семенов П. А., Процак А. М., Егоров В. П. — Цветная графика в микроЭВМ «Электроника 60» и «Электроника НЦ-80»	47
Микропроцессорная графическая станция ГТ-80	54
Алешин А. Н., Крюков С. Н. — Программное обеспечение символично-графического цветного телевизионного дисплея	56
Бабкин П. А., Солоненко О. П., Тарасов Б. В., Федорин В. Г. — Средства цветной машинной графики для микроЭВМ «Электроника 60»	60
Страбыкин Д. А. — Экспериментальные исследования микропроцессорных устройств с помощью диалоговых систем управления	62
Антоняк И. И., Зайшлый Я. С., Фриш А. Е. — Импульсный стабилизатор специализированных напряжений для микропроцессора КР580ВМ80	66
Поздеев В. С., Тихонов О. Н. — Отображение информации на дисплее ИЖВ	67
Чабан С. Д., Фильцагин Ю. А. — Модуль десятиканального аналогового вывода на базе микроЭВМ «Электроника 60»	70
Гнатьев Р. М., Скобылко А. Я. — Цифровой генератор синусоидальных сигналов с управлением по частоте на спецпроцессоре	71
Дианов А. П., Щелкунов Н. Н. — Организация динамической памяти микро-систем	75
Монахов В. Т. — Программы любителей для «Электроники БК-0010»	81
Водянкин А. Г., Моисеенко В. И. — Учебная локальная сеть микроЭВМ	83
Куправа Т. А. — Интерпретатор языка ФОКАЛ в микросхеме ПЗУ	84
Лапирев А. В., Рудометов Е. А., Харасов В. Г. — АЦП на БИС К1113ПВ1 для персональной ЭВМ «Электроника БК-0010»	85
Кулешова В. И. — Серийные микропроцессорные комплекты БИС	87

Главный редактор

А. П. ЕРШОВ

Редакционная
коллегия:

А. Г. Алексенко

В. М. Брябрин

А. А. Васенков

[зам. главного редактора]

И. Я. Вельбицкий

А. Б. Венгеров

Г. Р. Громов

[ответственный секретарь]

В. П. Иванников

М. Б. Игнатьев

А. В. Каляев

И. З. Карась

В. П. Куприянов

С. С. Лавров

В. В. Липаев

К. А. Меликан

И. А. Мизин

Б. Н. Наумов

[зам. главного редактора]

С. М. Пеленов

[зам. главного редактора]

А. К. Платонов

А. А. Попов

Д. А. Поспелов

Б. И. Рамеев

О. Л. Смирнов

А. А. Стогний

М. К. Сулим

Н. М. Шаруенко

Редакционный совет:

Р. Л. Ашастин

И. В. Бабянин

С. Н. Бушев

Е. П. Велихов

Н. Н. Говорун

В. В. Корчагин

В. П. Макаревич

И. И. Малашинин

А. Р. Назарьян

Ю. Е. Нестерихин

А. Л. Нефедкин

И. В. Прангишвили

Л. Н. Преснухин

В. В. Прихнялковский

Н. Л. Прохоров

Г. Г. Рябов

К. Н. Трофимов

В. И. Холлов

Н. Н. Шереметьевский

В. В. Шипльдин

А. В. Яковлев

Э. А. Якубайтис

Номер подготовили:

Е. И. Бабин, Г. Г. Глушкова,

В. М. Ларионова, С. С. Матвеев

Фото О. В. Чиркина

Адрес редакции журнала:

103051, Москва,

Малый Сухаревский пер., д. 9

Телефоны: 208-73-23, 208-19-94

Слано в набор 01.07.87.

Подписано к печати 27.08.87. Т-19003

Формат 84×108^{1/8}. Бумага № 1.

Высокая печать. Усл. печ. л. 10,08.

Уч.-изд. л. 14,8. Тираж 90 843 экз.

Знакз 310. Цена 1 р. 10 к.

Орган Государственного комитета

СССР по вычислительной технике

и информатике

Московская типография № 13

ПО «Периодика» ВО «Союзполи-

графиром» Госкомиздата СССР

107005, Москва, Девисовский пер.,

дом 30

ПЕРСОНАЛЬНЫЕ ЭВМ: КАК СДЕЛАТЬ
ИХ ПОЛЕЗНЫМИ

Начнем с парадоксальных наблюдений. Первое. Типичное письмо читателя: «Если читать только ваш журнал, то можно подумать, что ПЭВМ у нас на каждом углу. Но где они?». Второе. Я стараюсь внимательно следить за делами со школьной информатикой в СССР. И все же, когда в апреле этого года, готовясь к обзорному докладу по этому вопросу, попробовал прикинуть число ПЭВМ, установленных в школах, обнаружилось, что просчитался почти в два раза — в сторону занижения.

Вывод: хотя ПЭВМ еще практически нет, они появятся неожиданно и это может означать, что огромные затраченные средства останутся втуне, а что еще хуже, возлагаемые надежды на применение ЭВМ не оправдаются.

Нам нужно создать обстановку позитивного нетерпения в ожидании ПЭВМ. С нетерпением, по-видимому, все в порядке, но с позитивностью дело гораздо сложнее. Не будем пытаться затронуть все проблемы, поговорим о задачах, стоящих перед нашим журналом.

Скажем так: необходима убедительная реклама быстрой, эффективной и массовой полезности ЭВМ. К сожалению, в нашем общественном сознании понятие рекламы воспринимается как компрометирующий атрибут рыночной экономики. Сейчас становится очевидно, что в такой оценке мы с водой выплескиваем ребенка: самый лучший план становится реальностью не только и не столько тогда, когда изделие изготовлено и продано, но когда оно реализует свою потребительскую стоимость в руках пользователя.

Как массовый продукт ПЭВМ скорее ближе к автомашине, нежели к телевизору. Как и в случае автомобиля, успешное применение ПЭВМ требует воли и умения пользователя. И здесь делу может помочь только заразительность примера.

Мы крайне нуждаемся в хороших публикациях о применении ЭВМ, убедительно отвечающих на три главных вопроса: КАК НАЧАТЬ, КАК СДЕЛАНО, КАКОВ РЕЗУЛЬТАТ.

КАК НАЧАТЬ. Как ни странно, нам легче вообразить успешное применение ЭВМ в загоризонтный период полной информатизации, нежели в текущий предстартовый момент. В чем трудность начала? В необходимости запаса оптимизма, преодолющего трудности неприятия, инерции и незнания. В искусстве сочетания старых и новых форм работы, сокращения вынужденного периода ведения двойного производственного процесса: с компьютером и тут же, рядом без такового. В безошибочности выбора начальной точки приложения ЭВМ. И, конечно, в способности победить возможную неадекватность техники и программного обеспечения еще не познанным требованиям окончательного пользователя.

КАК СДЕЛАНО. Проблема состоит не в том, чтобы вместить в размеры статьи тысячу технических решений, всегда стоящих за любым применением ЭВМ. Здесь очень важно показать конкретную траекторию достижения цели, отобрать звенья цепи решений, попавших на критический путь; не упустить живой компонент в создании человеко-машинной системы. Главное в подаче и отборе материала — не авторский приоритет, не изобретательность или оригинальность, а убедительность в сочетании со 100-процентной честностью в описании трудностей и негативного опыта.

КАКОВ РЕЗУЛЬТАТ. У нас явный избыток повествовательно-констатационных публикаций без начала и конца. В результате вычислительная техника вырастает в самоцель, а ее пользовательский, орудийный аспекты уходят в тень, а это именно та сторона дела, где проваливается значительная часть компьютерных начинаний. Конечно, и конструктор, и схемотехник, и программист скажут: каждый должен делать свою часть, вот мой продукт, пользуйтесь на благо отечества. Хотелось бы еще раз подчеркнуть различие между пионерской работой и рутинным делом в сложившейся обстановке за гарантированное вознаграждение. Позволю себе привести приземленную метафору: прежде чем сифон начнет качать, необходимо протолкнуть жидкость собственным дыханием через всю трубку. Так и передовые работы по применению ЭВМ: любые технические решения имеют смысл, только если они хотя бы на шаг приближают нас к решению важнейших конечных задач научно-технического развития;

- производительность и качество труда,
- эффективность управления,
- экономия ресурсов,
- сохранение окружающей среды.

А. П. Ершов

УДК 681.327.11

Н. Н. Жуков, А. С. Коризенков, В. Н. Офицеров

ПЕЧАТАЮЩЕЕ УСТРОЙСТВО «ЭЛЕКТРОНИКА МС 6307»

Предназначено для работы в составе вычислительного комплекса (ВК) с ПЭВМ «Электроника МС 0585» или другими микроЭВМ, имеющими аналогичный интерфейс печати и протокол обмена.

Представляет собой последовательное матричное печатающее устройство ударного действия и может работать в режиме символьной печати, графическом режиме, режиме диагностики. Предусмотрены следующие виды печати: обычная, «качественная», печать «жирным» шрифтом, с подчеркиванием, печать символов двойной ширины. Возможны любые сочетания этих видов печати, за исключением того, что качественная печать не может сочетаться с печатью жирным шрифтом.

Печатающее устройство обладает средствами диагностики для проверки всех функциональных узлов, вплоть до выходных интерфейсных разъемов. Для этого предназначены встроенные автотесты проверки аппаратной части, проверки печати (обычной и качественной), автотест интерфейсов.

Среди режимов работы ПУ — ручное управление перемещением бумаги, программное управление форматом и видом печати, возможность получения информации о состоянии ПУ и ответов на запросы идентификации, возможность обмена по последовательному интерфейсу с различными параметрами обмена (протокол, скорость, разрядность кодов, наличие и вид контроля паритета).

Функциональные возможности ПУ могут быть реализованы пользователем программно (путемсылки различных управляющих кодовых последовательностей) и с помощью модульного переключателя, установленного на блоке контролера.

Технические характеристики печатающего устройства «Электроника МС 6307»

Формат бумаги, мм	220
ширина	до 100
диаметр рулона	Стык С2, ГОСТ 18145-81
Интерфейсы	(выходные уровни ± 6 В); ИРПС, ОСТ 11.305.916-84 (20 мА токовая петля) 75...4800
Скорость обмена по интерфейсам, Бол	20
Длина интерфейсного кабеля, м, не более	255
Объем входного буфера, байт	КОИ7, КОИ8
Коды	80
Число символов в строке	40
Скорость печати, знаков/с	5×9
Размер матрицы, точек	9×9
обычная печать	0,192
качественная печать	0,384
Разрешающая способность, мм	960
по горизонтали	5...40
по вертикали	360×320×180
Число точек в строке при печати в графическом режиме, не менее	100
Диапазон рабочих температур, °С	
Габаритные размеры, мм	
Потребляемая мощность, Вт, не более	

Предусмотрены протоколы обмена управляющими кодами «включить обмен», «отключить обмен» (XON/XOFF) и потенциальным сигналом «готовность» (READY/BUSY). Набор символов составляют национальные КОИ-7Н1, КОИ-7Н2, специальный национальный, псевдографика VT100, код КОИ-7Н0.

Печатающее устройство содержит три основных функциональных узла (см. рисунок): микроЭВМ, блок управления электроприводами, печатающий механизм.

МикроЭВМ построена на базе микропроцессорного набора КР580 и имеет следующие основные характеристики:

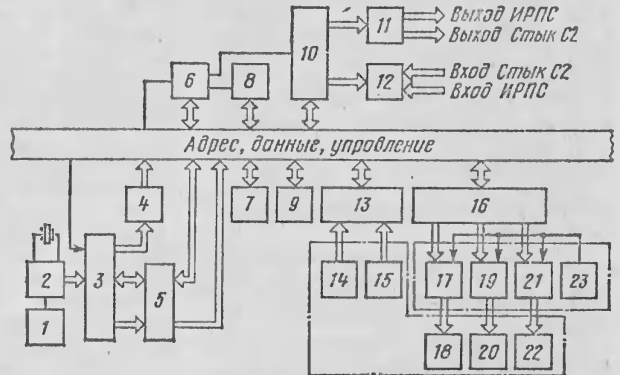
Тактовая частота, МГц	2
Объем ОЗУ, К байт	4
Объем ПЗУ, К байт	10
Число уровней прерывания	2
Число управляющих входов	8
Число управляющих выходов	20
Число интерфейсов	2

По управляющим входам в микроЭВМ поступает информация с пульта управления ПУ и датчиков печатающего механизма.

Выходы микроЭВМ управляют печатающей головкой (ПГ), шаговыми двигателями (ШД) каретки и вала печатающего механизма. Необходимые для работы ПУ временные интервалы задаются программируемым таймером, имеющим высший уровень приоритета прерывания. Низший уровень имеет порт последовательного ввода-вывода, через который происходит обмен с управляющей ЭВМ по интерфейсу ИРПС или Стык С2.

Коды, поступающие от управляющей ЭВМ, по прерываниям записываются во входной буфер ПУ. Когда число кодов в буфере становится больше 128, ПУ инициирует запрещение обмена. Если ПУ находится в режиме печати, то производится выборка кодов из входного буфера и их отработка. Когда число кодов в буфере становится меньше 32, обмен с ЭВМ возобновляется по инициативе ПУ.

Коды графических символов из входного буфера переписываются в буфер выходной строки, формирование которого продолжается либо до получения управляющего кода, определяющего конец строки, либо до фор-



Структурная схема печатающего устройства «Электроника МС 6307»:

1 — цепь начальной установки; 2 — генератор КР580ГФ24; 3 — микропроцессор КР580ИК80; 4 — буфер адреса; 5 — системный контроллер КР580ВК28; 6 — контроллер прерываний КР580ВН59; 7 — ОЗУ; 8 — таймер КР580ВН53; 9 — ПЗУ; 10 — контроллер последовательного ввода-вывода КР580ИКБ1; 11 — выходные формирователи; 12 — входные формирователи; 13, 16 — контроллеры параллельного ввода-вывода КР580ИК55; 14 — переключатели режимов работы; 15 — пульт управления; 17 — схема управления ШД вала; 18 — ШД вала; 19 — схема управления ШД каретки; 20 — ШД каретки; 21 — схема управления ПГ; 22 — ПГ; 23 — схема защиты

мирования числа кодов, равного длине строки. Далее буфер выходной строки преобразуется в очередь, представляющую собой последовательность элементов, каждый из которых определяет некоторое воздействие на исполнительные механизмы ПУ. Управляющие символы обрабатываются по мере выборки их из входного буфера (соответствующие управляющие воздействия также помещаются в очередь).

С момента появления в очереди первого элемента производится его обработка, т. е. непосредственно процесс печати. Процессы формирования входного буфера, выходной строки и очереди, обработка очереди протекают псевдопараллельно.

Печатающая головка имеет 9 игл, расположенных по вертикали. Каждый печатаемый символ отображается в виде матрицы. Печать символов осуществляется при дви-

жении ПГ в обе стороны. В графическом режиме печать заполняется шестью иглами при движении ПГ в одну сторону.

Блок управления электроприводом выполнен в основном на дискретных элементах и служит для непосредственного управления ПГ и ШД от мощных источников тока. В блоке реализована защита ПГ от случайных отказов схемы управления ПГ и силовых цепей питания.

С пульта управления оператор может перемещать бумагу на одну строку (или страницу) вперед и назад, останавливать и включать процесс печати, запускать эталонные, контролируемые работу ПУ.

Статья поступила 21 октября 1986 г.

УДК 681.325.5 : 681.327.8.06

Р. Р. Габдуллин, В. В. Горемыкин, А. П. Горяшко, Л. С. Косов,
Г. А. Лукович, В. Г. Мионов

КОМПЛЕКТ БИС ДЛЯ ВСТРОЕННОГО ДИАГНОСТИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Общие принципы самотестирования

Постоянно растущая функциональная сложность интегральных микросхем и соответственно вычислительных и управляющих систем на их основе чрезвычайно обострила проблему тестового диагностирования на всех этапах жизненного цикла системы [1].

Большинство ведущих разработчиков сегодня уже понимает, что решение проблем тестового диагностирования будет в конечном счете определять возможность успешного серийного производства и эксплуатации системы. Эта зависимость становится тем заметнее, чем выше тиражность системы (например, для персональных ЭВМ). Однако от осознания до активного изменения технической политики зачастую проходит большой срок. Одна из объективных причин такого запаздывания — отсутствие подходящей элементной базы, позволяющей реализовать перспективные методы проектирования тестопригодной аппаратуры с приемлемыми на практике затратами.

Можно условно выделить три группы методов тестопригодного проектирования:

проектирование тестопригодных устройств с помощью какого-либо способа сканирования внутренних состояний [2];

проектирование самотестируемого устройства с помощью встраивания (на уровне БИС или на уровне сменных модулей) дополнительных схем, осуществляющих автономный процесс тестового диагностирования [3];

сочетание сканирования с методами самотестирования (примером реализация может служить модуль TCM фирмы IBM [4]).

Сегодня, по-видимому, не существует четких рекомендаций относитель-

но того, какому из предлагаемых методов и в какой ситуации следует отдавать предпочтение. Многие ведущие зарубежные фирмы пробуют все альтернативные варианты. Ясно одно: необходимое условие жизнеспособности метода тестопригодного проектирования зависит от выбора подходящей элементной базы. Например, эффективная реализация сканирования для организации сканирующего пути требует применения Д-триггеров или специальных триггеров с дополнительным установочным входом для организации режима диагностирования.

Организация режима самотестирования особенно целесообразна при проектировании с использованием базовых матричных кристаллов (БМК) [5, 6]. В этих случаях вводится специальное расширение библиотеки функциональных элементов, для проектирования схем на БМК.

Решение проблем тестового диагностирования для микропроцессорных устройств длительное время полностью возлагалось на внешние средства — автоматизированные стенды контроля. Однако методы внешнего тестового диагностирования не могут «угнаться» за все возрастающей сложностью и насыщенностью микропроцессорных плат, поэтому были сделаны попытки изменения структуры микропроцессоров.

Структура микропроцессора модифицируется таким образом, чтобы непосредственно, без выполнения команд микропроцессора, производить запись и чтение всех регистров. При этом большая часть регистров микропроцессора и логики устройств управления становится доступной для непосредственного наблюдения в режиме тестового диагностирования. Стоимость такой модификации (в числе

вентилей) составляет примерно 10% [7].

Подобная модификация позволяет обнаружить более 98% одиночных константных неисправностей с помощью теста, содержащего 4700 входных наборов [8].

В последнее время появились также сообщения о модификациях структуры серийно выпускаемых микропроцессоров [9]. Например, введены дополнительные схемы в 16-разрядный МП M68000, позволяющие уменьшить время проверки в 60...100 раз. Средствами самотестирования снабжается также 32-разрядный микропроцессор MC68020, его вспомогательный арифметический сопроцессор MC68881 и микропроцессор 80386 фирмы Intel [5].

Средства самотестирования в этих кристаллах ориентированы в первую очередь на автономную проверку программируемых логических матриц управляющей логики микропроцессора, так как они не наблюдаются непосредственно через входные контакты кристалла и поэтому крайне сложно поддаются проверке с помощью внешних средств тестового диагностирования.

Модификация структуры МП облегчает диагностирование только микропроцессора и не решает проблем тестового диагностирования микропроцессорных систем, содержащих БИС ОЗУ, интерфейсные БИС в сложную управляющую логику, реализованную с помощью СИС.

Единственная, известная авторам, попытка облегчить процесс диагностирования микропроцессорных систем предпринята фирмой AMD, выпустившей микросхему AM 29818 («последовательный теневого регистр») [10]. Эта микросхема представляет собой два 8-разрядных параллельно-последовательных регистра, один из которых используется как рабочий конвейерный регистр, а второй позволяет следить за состоянием рабочего регистра и устанавливать его в требуемое состояние в режиме диагностирования. Наличие теневого регистра помогает проектировать тестопригодные микропроцессорные устройства, однако не избавляет от необходимости иметь сложное и дорогостоящее стен-

довое оборудование на этапе производства и эксплуатации системы. По-видимому, единственный способ существенно ослабить требования к стендовому оборудованию и упростить процесс восстановления сложной микропроцессорной системы состоит в переходе к проектированию самотестируемых микропроцессорных плат, т. е. максимально автономному режиму тестового диагностирования.

В соответствии с общей моделью процесса самотестирования [3] организация автономного процесса тестового диагностирования предполагает наличие схем, задающих тестовые воздействия, и схем, анализирующих реакции диагностируемого устройства. При выборе конкретного способа генерации воздействий и анализа реакций приходится искать компромиссные варианты между просто реализуемыми, но не всегда гарантирующими требуемую полноту обнаружения, методами компактного тестирования и традиционными способами тестирования и традиционными способами тестового диагностирования с помощью заранее составленных тестовых программ. Обычно компромисс достигается при сочетании небольших по объему тестовых программ, ориентированных на проведение установочных экспериментов, и генерации достаточно длинных псевдослучайных последовательностей (2^{16} и более), причем в качестве методов анализа используется арифметическое суммирование или сигнатурное сжатие. Такой состав дополнительного оборудования позволяет организовать самотестирование сложных устройств, реализованных на базе однокристалльных или секционных МП.

Диагностирование ОЗУ требует реализации метода самотестирования. Широкий класс неисправностей ОЗУ с произвольной выборкой (за исключением неисправностей типа взаимного влияния ячеек накопителя) обнаруживается алгоритмическими тестами с линейными оценками длины. Экономичная реализация самотестирования плат ОЗУ возможна лишь при условии, что будут аппаратно реализованы подходящие алгоритмические тесты проверки.

Таким образом, к числу обязательных функций, которые необходимы при реализации любого способа встроенного самотестирования, следует отнести:

- генерацию псевдослучайных векторов;
- реализацию параллельного и(или) последовательного сигнатурного сжатия, арифметического суммирования, синдромного сжатия (подсчет числа единиц, переходов и т. п.);
- реализацию одного или нескольких алгоритмов проверки ОЗУ.

Экспериментальные результаты показывают, что реализация этих функций на элементах серии К533 при организации метода самотестирования для типичных структур микроЭВМ и

плат ОЗУ емкостью 64К слов приводит к избыточности 18...20% [11]. Для большинства разработок такая избыточность оказывается неприемлемой. Единственный способ, позволяющий по крайней мере вдвое уменьшить требуемую избыточность, связан с применением заказных БИС, ориентированных на задачи самотестирования (так же как микросхема АМ 29818 ориентирована на метод диагностирования под названием SSR, т. е. последовательный теневой регистр).

Комплект БИС для встроенных средств контроля

Комплект БИС серии КР1828 предназначен для тестового диагностирования процессоров, устройств памяти, процессоров ввода-вывода, схем произвольной логики и т. д. Интегральная реализация тест-структур позволяет конструктивно объединять их с объектами контроля и осуществлять проверку по принципу «годен — не годен» на реальных частотах работы устройств, обеспечивая эффективное обнаружение не только константных неисправностей и замыканий, но и параметрических отказов.

Наибольший эффект достигается при встраивании тест-структур в каждый типовой элемент замены (ТЭЗ), что позволяет быстро обнаруживать и заменять отказавший ТЭЗ. Кроме того, тест-структура может быть использована для поиска неисправности в ТЭЗе, исключая необходимость в применении сложных и дорогостоящих стенов.

Комплект включает две БИС, изготовленные по ТТЛШ-технологии и совместимые по уровням сигналов и номиналу питающего напряжения с широко распространенными семействами ИС. Обе БИС построены по разрядномодульному принципу, что позволяет практически неограниченно увеличивать длительность формируемых и анализируемых сигналов.

БИС КР1828ВЖ1 (рис. 1) предназначена для построения ядра схем встроенной диагностики процессоров различного типа, ПЗУ/ППЗУ и схем произвольной логики.

Устройство управления на основе заданной команды и внешних сигнала

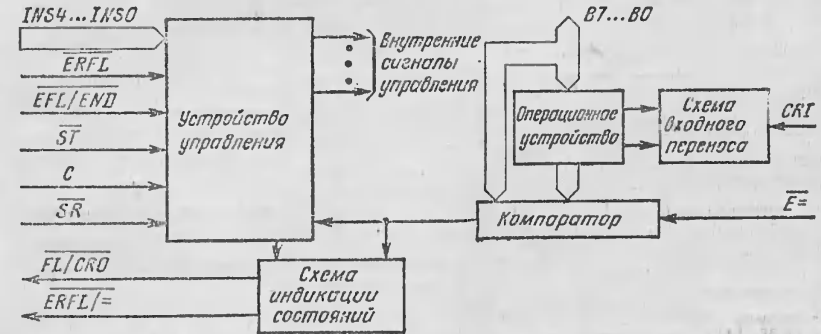


Рис. 1. Структурная схема БИС КР1828ВЖ1

лов формирует необходимые сигналы для работы всех узлов БИС. Операционное устройство содержит сумматор с аккумуляторами и выполняет все необходимые преобразования получаемых данных. Компаратор сравнивает полученные результаты с эталонными значениями, хранящимися во внешнем ПЗУ. Устройство входного переноса обеспечивает работу операционного устройства в режимах сигнатурного анализа. Узел индикации предназначен для формирования результата тестирования в виде сигналов Да и Нет и секционного наращивания БИС. Условное графическое обозначение БИС показано на рис. 2, назначение выводов приведено в табл. 1.

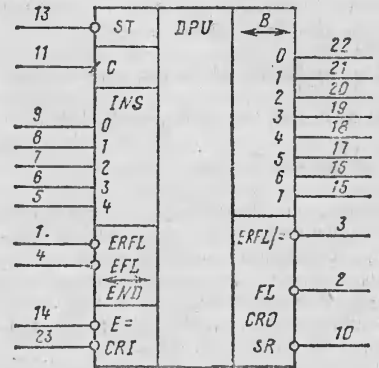


Рис. 2. Условное графическое обозначение БИС КР1828ВЖ1

БИС может функционировать в одном из следующих режимов:

- генератор псевдослучайных кодов с максимальным периодом 255;
- генератор инкрементируемых и декрементируемых последовательностей (от 0 до 255 и обратно);
- сигнатурный анализатор с (или без) сравнением(я) с эталоном;
- счетчик числа переходов со (или без) сравнением(я) с эталоном;
- сумматор со сравнением полученной и контрольной сумм.

Основные технические характеристики БИС КР1828ВЖ1

Разрядность (с возможностью наращивания) 8

Минимальная длительность такта, нс, не более	200
Максимальный ток потребления, мА, не более	260
Тип корпуса	239.24-2
Диапазон рабочих температур, °С	-10...+07

Назначение выводов БИС КР1828ВЖ1

Вывод	Обозначение	Назначение
1	EERFL	Вход управления. Сообщение о формировании последнего тестового воздействия
2	FL/CR0	Выход переноса или сигнала о положительном результате тестирования (Да). Два состояния
3	ERFL/EO	Выход схемы сравнения или сигнал об отрицательном результате тестирования (Нет). Два состояния
4	EFL/CR	Вход разрешения формирования сигналов Да, Нет или выход переноса сумматора (в режиме сумматора)
5...9	INS4...INS0	Вход команды
10	SR	Выход начальной установки. Два состояния
11	C	Вход тактовых импульсов
13	ST	Вход включения микросхемы в рабочий режим
14	EI	Вход разрешения сравнения
15...22	B7...B0	Двухнаправленная магистраль. Выдача данных и прием эталонов
23	CR1	Вход переноса сумматора. В режимах сигнатурного анализатора и счетчика числа переходов — вход анализируемого сигнала
12	OB	Общий
24	U _{cc} 5B	Питание

Номинал питающего напряжения и уровни сигналов соответствуют номиналу и уровням ТТЛ-схем.

БИС КР1828ВЖ2 предназначена для применения в качестве ядра построения схемы встроенной диагностики ОЗУ. Многообразие проверяющих тестов, обусловленное схемотехническими и конструктивными (топологическими) особенностями создания различных типов БИС ОЗУ, свидетельствует о том, что для современных БИС ОЗУ, даже при ограничениях на модель неисправностей, не существует универсального проверяющего теста практически приемлемой длины. Положение еще более осложняется, если объектом контроля служит ячейка, содержащая несколько десятков БИС ОЗУ.

Кроме практических ограничений на длительность проведения тестового эксперимента необходимо также учитывать схемотехнические особенности аппаратной реализации теста как в виде одной БИС, так и при организации совместной работы нескольких БИС.

Для аппаратной реализации в виде БИС КР1828ВЖ2 (рис. 3) выбраны следующие тесты:

- обращение по прямому и дополнительно адресам (MASEST);
- пропорциональное деление;
- бегущие по разрядам «1» и «0».

Первые два теста предназначены для проверки накопителя и схем дешифрации ОЗУ, последний — для диагностики информационного тракта (выявления замыканий разрядных цепей). Суммарная длина тестов 40п, где п — число ячеек ОЗУ.

Условное графическое обозначение микросхемы приведено на рис. 4. Назначение выводов — в табл. 2.

Основные технические характеристики БИС КР1828ВЖ2

Разрядность (с возможностью наращивания) данных	8
адреса	8
Минимальная длительность такта, нс, не более	200
Максимальный ток потребления, мА, не более	300
Тип корпуса	244.48-5
Диапазон рабочих температур, °С	-10...+70

БИС серии КР1828 могут быть использованы в разнообразных стационарных и переносных стендах контроля, а также для построения ручных и настольных сигнатурных анализаторов, счетчиков различного типа и другой аппаратуры.

Применение комплекта БИС серии КР1828 в микропроцессорных системах

Схемы самотестирования, основу которых составляют БИС КР1828ВЖ1 и КР1828ВЖ2, целесообразно размещать на сменных узлах (платах) микропроцессорных систем,

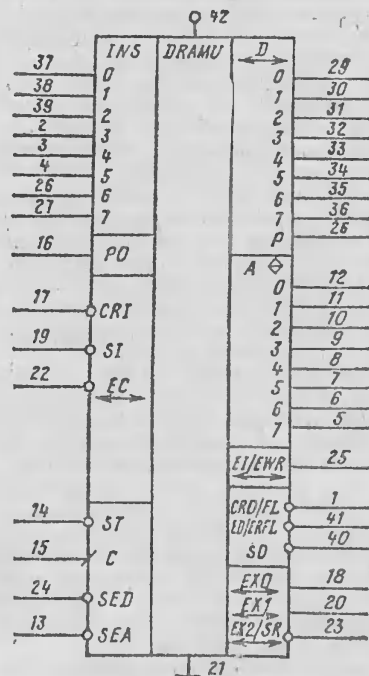


Рис. 4. Условное графическое обозначение БИС КР1828ВЖ2

В дальнейшем эти БИС будем называть ТЭДИ1 и ТЭДИ2 (БИС Тестовой Экспрессе Диагностики). Схемы самотестирования (иногда называемые тест-структурами) по сигналу с инженерного пульта осуществляют тестовое диагностирование сменного узла, в котором они размещены, на рабочей частоте по принципу «годен — не годен».

Для процессора, имеющего стандартную архитектуру процессоров и контроллеров с микропрограммным

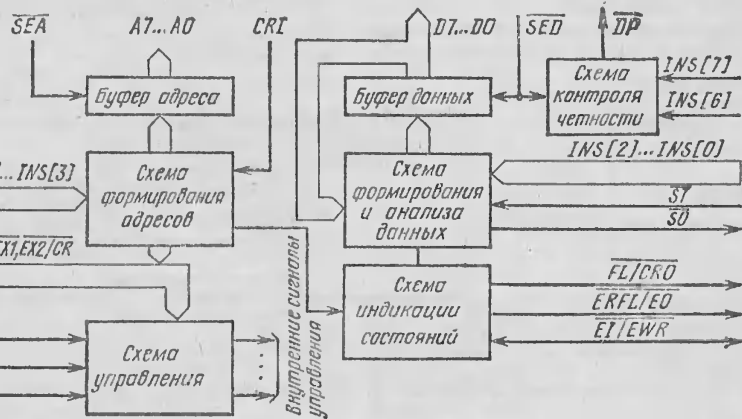


Рис. 3. Структурная схема БИС КР1828ВЖ2

Назначение выводов БИС КР1523ВЖ2

Выход	Обозначение	Назначение
2, 1, 48...43	A0...A7	Шина адреса. Выход с тремя состояниями
3	SEA	Вход разрешения выдачи адреса
4	ST	Вход включения микросхемы в рабочий режим
5	C	Вход тактовых импульсов
6, 7	PI, P0	Вход выбора позиции
8	CR1	Вход переноса адреса
9	EXO	Вход-выход объединения микросхем
10	SI	Вход сдвига влево
11	EX1	Вход-выход объединения микросхем
14	EC	Вход-выход управления счетчиком циклов
15	EX2/SR	Вход расширения. Выход начальной установки
16	EI/EWR	Вход разрешения сравнения. Выход сигнала ЗП/ЧТ
17	SED	Вход разрешения выдачи данных
19, 20, 18, 35	INS6...INS9	Вход команды управления битом четности
21	DP	Вход-выход бита четности (нечетности)
22...30	D0...D7	Вход-выход шины данных
31...33	INS0...INS2	Вход команды управления разрядностью данных
34	SO	Выход сдвига влево
36	ERFL/E0	Выход схемы сравнения или сигнал отрицательного результата тестирования (Нет)
39	FL/CRO	Выход переноса адреса или сигнал положительного окончания тестирования (Да)
40...42 12, 13, 24	INS3...INS5	Вход команды управления разрядностью адреса
37, 38	OB CC, 5B	Общий Питание

управлением (рис. 5), может быть использована тест-структура, показанная на рис. 6. Генератор тестовых воздействий формирует 24-разрядный тестовый вектор (микрокоманду процессора). Поля микрокоманды с расширенными кодами формируются с помощью ПЗУ, которое может адресоваться генератором псевдослучайных чисел (ГПСЧ) или специальным счетчиком. Максимальное число тестовых векторов (Т), формируемых ГПСЧ, при использовании двух схем ТЭДИ1, равно 64770. Обязательное условие при тестировании — установка процессора в некоторое заранее известное состояние (установочный эксперимент).

Анализ реакций на тестовые воздействия осуществляется при регистрации значений сигналов в N заранее выбранных контрольных точках на всех тестовых воздействиях. Получаемая в контрольной точке T-разрядная последовательность превращается в процессе эксперимента в 8-разрядную сигнатуру, которая сравнивается со значением эталонной сигнатуры (рис. 7).

Схема анализа реакций состоит из 32-разрядного коммутатора контрольных точек, счетчика на основе БИС ТЭДИ1, ПЗУ эталонных сигнатур КР556РТ5 и 8-разрядного сигнатурного анализатора на БИС ТЭДИ1. Таким образом показанная на рис. 6 схема 32 раза формирует тестовую последовательность и сравнивает эталонные 8-разрядные реакции с полученными в каждой из 32 контрольных точек,

Схема управления тест-структурой состоит из двух вентилях и триггера. Функциональная схема модуля ОЗУ, построенного на основе БИС КР541РУ3, с тест-структурой на двух БИС ТЭДИ2 показана на рис. 8. Емкость модуля ОЗУ — 64К 18-разрядных слов (16 информационных и два контрольных разряда). Для такого модуля введение тест-структуры приводит к 10%-ной избыточности в пересчете на суммарное число выводов микросхем.

Функциональная схема модуля ПЗУ с тест-структурой на основе че-

тырех БИС ТЭДИ1 приведена на рис. 9. Тест-структура состоит из счетчика адресов, сумматора со сравнением, схемы управления.

Счетчик адресов реализован на двух БИС ТЭДИ1, включенных в режиме генератора инкрементных состояний. Сумматор со сравнением реализован на двух БИС ТЭДИ1, включенных в режиме сумматора со сравнением и учетом переноса из старшего разряда. Схема сопряжения с магистралью реализована на двух триггерах и трех вентилях. Она формирует управляющие сигналы для шины микроЭВМ и управляет работой тест-структуры. Для конкретного модуля ПЗУ на основе БИС КР556РТ7 емкостью 24К 16-разрядных слов введение тест-структуры приводит к 10%-ной избыточности.

Возможности методов встроенного диагностирования в плане обнаружения неисправностей оценивались экспериментально в процессе физического моделирования. Для конкретного процессора, на котором осуществлялись экспериментальные исследования, тест-структура, показанная на рис. 6, требует 9% всего оборудования в пересчете на суммарное число выводов микросхем и менее 5% в пересчете на корпус.

В процессе эксперимента оценивалась прежде всего зависимость полноты обнаружения заданного класса неисправностей от длины тестовой последовательности и от числа контрольных точек. Рассмотренный класс неисправностей — константные неисправности на выводах ИС. Из 739 целей процессора оказались непроверяемыми 95. Это связано с тем, что процессор проектировался без учета требований самотестирования.

Все проверяемые цели были разбиты на 152 класса эквивалентности. В каждом классе эквивалентности выби-

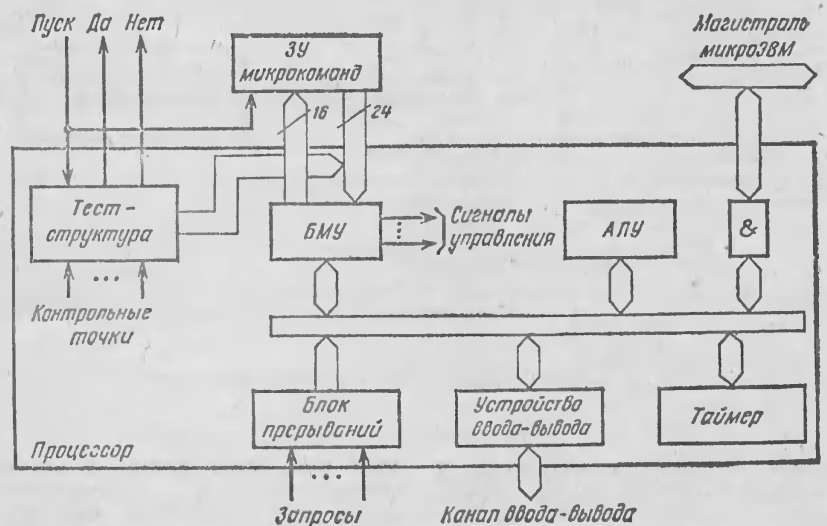


Рис. 5. Структурная схема процессора

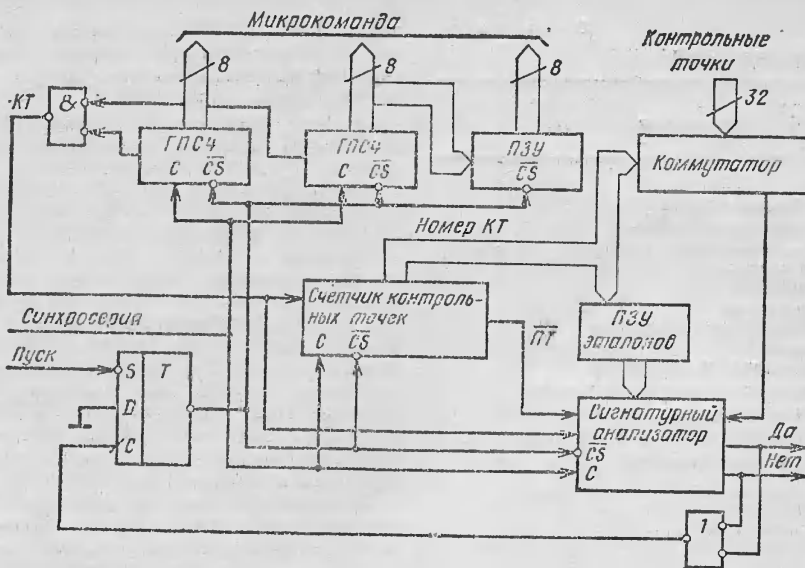


Рис. 6. Схема тест-структуры процессора

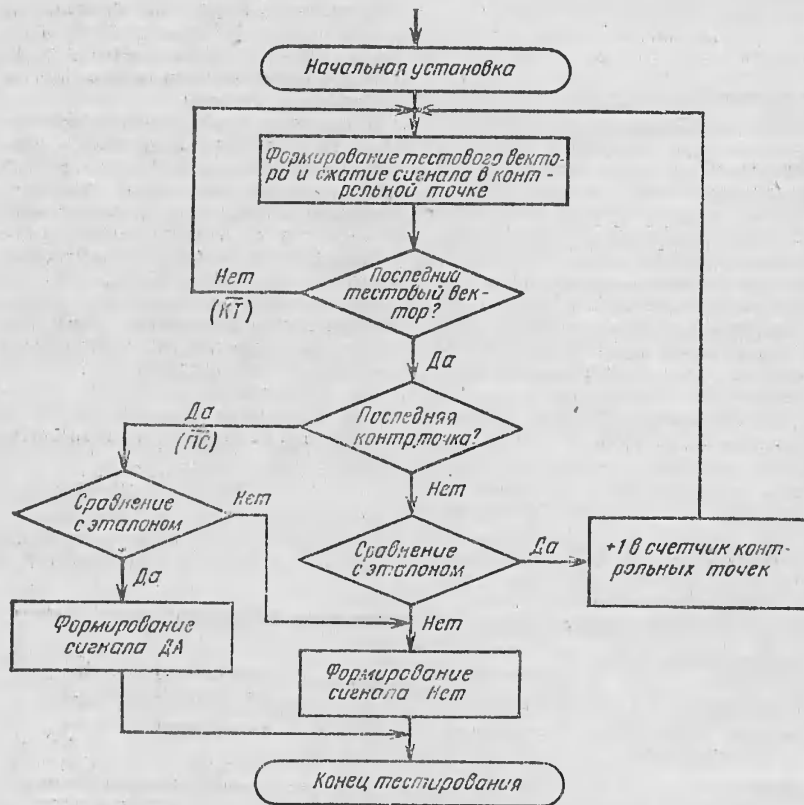


Рис. 7. Алгоритм работы тест-структуры процессора

исправностей от длины эксперимента и выбранного числа контрольных точек.

Эффективность рассматриваемых способов тестового диагностирования зависит не только от метода самотестирования и его схемной реализации, но и от уровня «приспособленности» системы к проведению самотестирования. Эксперименты по организации режима самотестирования в микропроцессорных системах позволили сформулировать ряд требований, без учета которых на этапе проектирования системы трудно рассчитывать на высокую эффективность предлагаемых методов.

1. Необходимость в процессе тестового эксперимента разрыва обратных связей, которые могут привести к неоднозначности при смене состояний устройства под воздействием тестовой последовательности.

2. Синхронизация всех асинхронных схем (контроллеры регенерации динамических ОЗУ, схемы обработки прерываний, каналы прямого доступа).

3. Возможность перевода работы времязадающих схем (программируемые таймеры, ждущие мультивибраторы и т. п.) на сокращенные временные интервалы.

4. Уменьшение числа фронтов синхросигналов, по которым происходит изменение логических уровней диагностируемой схемы.

5. Исключение неоднозначных уровней сигналов, которые могут формироваться схемами с тремя состояниями или выходами схем с открытым коллектором (эмиттером).

Выполнение перечисленных требований позволяет обнаруживать не только терминальные неисправности микропроцессорных плат, но и широкий класс параметрических отказов, поскольку диагностирование осуществляется на рабочей частоте. Кроме того, существенно упрощается поиск неисправности (с точностью до неисправной цепи). Частично такая информация может быть получена из номера контрольной точки, по которой произошло несовпадение с эталоном. Для получения более точной информации следует организовать поиск неисправностей с помощью щупа, в котором совмещен сигнатурный анализатор и счетчик числа переходов. Такой щуп, если в нем обеспечена возможность работы от среза и фронта любой используемой синхросерии и формирование строба работы, позволяет достаточно быстро находить неисправности в платах, насчитывающих 100...180 микросхем.

Телефон для справок 273-99-01, Москва.

ралось по одному полюсу схемы. В выбранный полюс вносились физические неисправности $\equiv 0$, $\equiv 1$. Эксперимент проводился на частоте 4 МГц на пяти различных тестовых последовательностях ГПСЧ: 403,

2077, 8777, 32881, 64770. В случае последовательности максимальной длины и 32 контрольных точек время эксперимента составляло не более 2 с. На рис. 10 показана зависимость процента обнаруженных не-

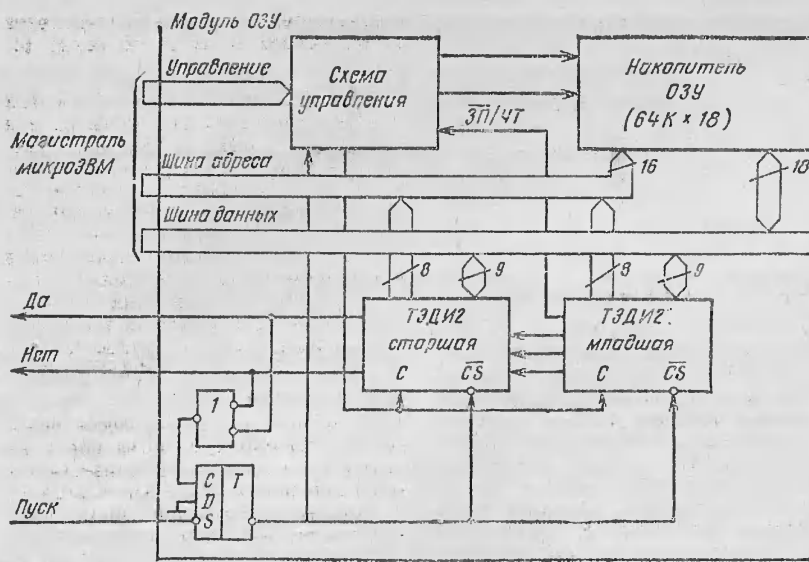
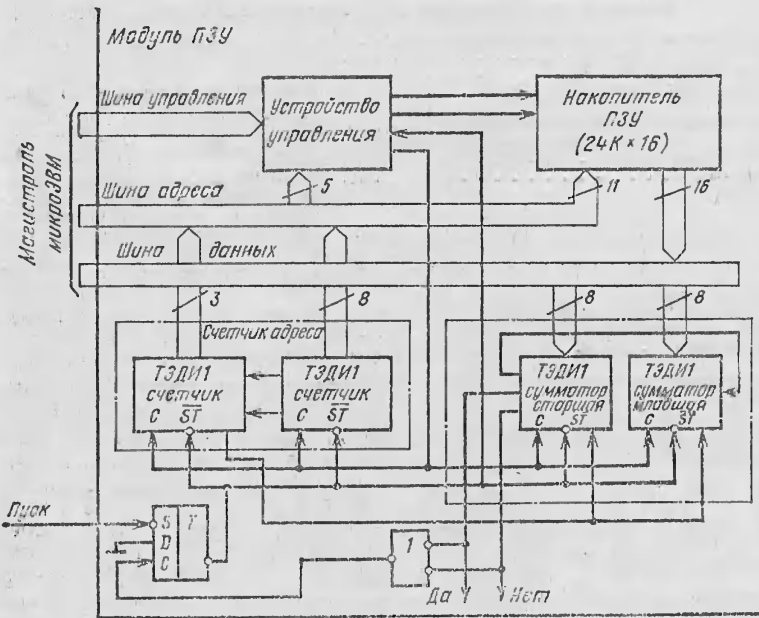


Рис. 8. Функциональная схема модуля ОЗУ

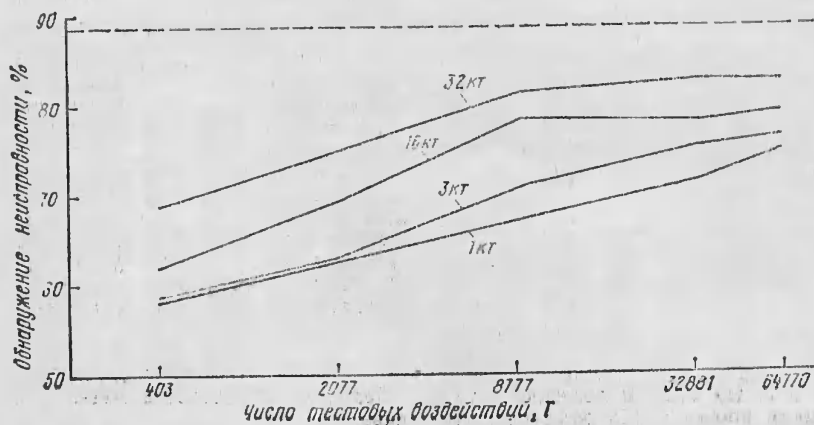
Рис. 9. Функциональная схема модуля ПЗУ

Рис. 10. Зависимость обнаруживающей способности тест-структуры от длины тестовой последовательности и числа контрольных точек



ЛИТЕРАТУРА

1. Борисов В. С., Горяшко А. П. Методы встроенного диагностирования микропроцессорных средств вычислительной техники // Микропроцессорные средства и системы.—1984.—№ 2.—С. 36—41.
2. Горяшко А. П. Синтез легко тестируемых схем: идеи, методы, реализация // Автоматика и телемеханика.—1984.—№ 7.—С. 5—35.
3. Системы тестового контроля современных ЭВМ / А. П. Горяшко, И. Н. Литвиненко, Л. В. Поспелов и др. // Изв. АН СССР. Техническая кибернетика.—1985.—№ 1.—С. 37—56.
4. Bardell P. H., Mc Aneeey W. H. Self-testing of Multichip Logic Modules // Int. Test Conf.—1982.—Nov. 15—18.—P. 200—205.
5. Мендельсон А. Постепенное развитие разработок ИС с самодиагностированием // Электроника.—1986.—Т. 59.—№ 4.—С. 46—53.
6. Испытание сложных СБИС с помощью стандартного оборудования // Электроника.—1986.—Т. 59.—№ 11.—С. 39—45.
7. Parthasarathy R., Reddy S. M., Kuhl J. G. A Testable Design of General Purpose Microprocessors // 12-th Int. Symp. Fault Tolerant Comput.—1982.—June.—P. 117—124.
8. Nanda S., Reddy S. M. Design of Easily Testable Microprocessors—a Case Study // Int. Test Conf.—1982.—Nov. 15—18.—P. 480—484.
9. Muehldorf E. J., Savkar A. D. LSJ Logic Testing on Overview // IEEE Trans. Comput.—1981.—Jan.—P. 81—94.
10. Lee F., Coli V., Muller W. On-drip System's Logic States // Electronic Des.—1983.—Vol. 31.—№ 8.—April 14.—P. 121—128.
11. Синтез легко тестируемых дискретных устройств четвертого поколения / А. М. Бойкович, А. П. Горяшко, В. Г. Мионов и др. // III Междуна. симп. Техническая диагностика ИМЕКО.—М.: Ин-т проблем управления.—1983.—С. 40—42.



Статья поступила 25 декабря 1986 г.

УДК 661.321—181.48

Л. М. Лемко, В. В. Гладков, С. В. Ермаков, В. Н. Жуков

ПЕРСОНАЛЬНЫЙ МИКРОКОМПЬЮТЕР «ЭЛЕКТРОНИКА МК 85»

«Электроника МК 85» представляет собой миниатюрный персональный компьютер, (ПК), внешне похожий на обычный карманный микрокалькулятор (рис. 1). Он может работать как в режиме калькулятора, так и в режиме ЭВМ с использованием версии языка Бейсик. Ввод информации и управление «Электроникой МК 85» осуществляются с клавиатуры, состоящей из 54 клавиш, расположенных в двух зонах. В левой зоне размещены 35 многофункциональных клавиш, предназначенных для ввода в ПК прописных или строчных букв латинского и русского алфавита, математических и специальных знаков, команд и операторов языка Бейсик, а также для управления курсором и выбора режима работы как самого ПК, так и его клавиш.

В правой зоне находятся 19 одно-, двух- или трехфункциональных клавиш, с помощью которых вводятся цифры, несколько букв русского алфавита, выбирается нужный файл оперативной памяти, а также осуществляются некоторые функции управления компьютером.

Многофункциональность (до семи различных функций) большинства клавиш достигается за счет наличия клавиш совмещенных функций S и F, а также клавиш MODE (выбор режима). Выполняемые функции обозначены на самих клавишах, сверху, снизу и справа от них, а также на специальной наклейке на клавиатуру.

Использование принципа «бегущей» строки позволяет записывать в ПК строки длиной до 63 символов. На жидкокристаллическом матричном 12-разрядном индикаторе с регулируемой контрастностью одновременно отображаются до 12 букв, цифр или символов. При помощи клавиш перемещения курсора ← и → можно просмотреть строку любой длины. В верхней части индикатора расположена служебная строка, в которой индицируются символы, обозначающие режим работы компьютера и число неиспользованных шагов программы (рис. 2).

Индикатор работает в мультиплексном режиме со степенно мультиплексированием 1:16. Такой матричный режим адресования по сравнению со статическим режимом, когда каждый элемент индикатора

управляется индивидуально, позволяет значительно сократить число необходимых управляющих электродов. Основной принцип такого адресования состоит в том, что на строки X1, X2, ..., XN матрицы N×M последовательно, через равные промежутки времени T/N, подаются однополярные импульсы с амплитудой U1. Одновременно на столбцы Y1, Y2, ..., YM подаются записывающие

импульсы с амплитудой ±U2. Элемент матрицы переходит во включенное состояние (выбранное), если на него подано результирующее напряжение U0=U1+U2, и остается включенным (полувыбранное состояние) при напряжении U1—U2.

Максимальное число шагов программы — 1221, что позволяет вводить в компьютер в среднем до 150 строк программы на Бейсике.

Основные характеристики ПК «Электроника МК 85»

Система счисления при вводе и выводе данных	десятичная
Число разрядов мантиссы	10
Число разрядов порядка	4
Число адресуемых регистров памяти	26 (с возможностью расширения до 178)
Объем энергонезависимой памяти, байт	2К
Число одновременно хранимых программ (файлов)	10 (P0 ... P9)
Язык программирования	Бейсик
Режим работы	основной, калькуляторный, запись, отладки, совмещенных функций S и F, расширения функциональных возможностей, повышенного быстродействия
Питание	4 элемента СЦ 0,18 или блок питания
Потребляемая мощность, Вт, не более	0,02
Диапазон рабочих температур, °С	5...40
Габаритные размеры, мм	13×165×73
Масса, г, не более	150

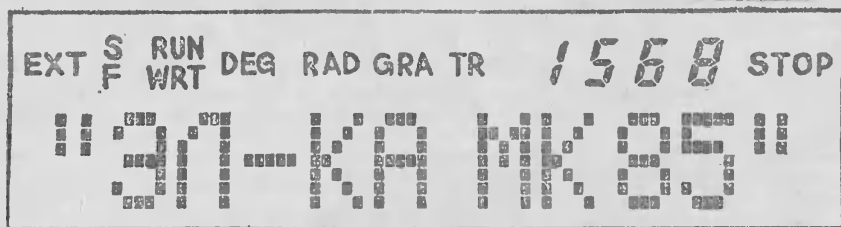


Рис. 1. Верхняя панель ПК «Электроника МК 85»

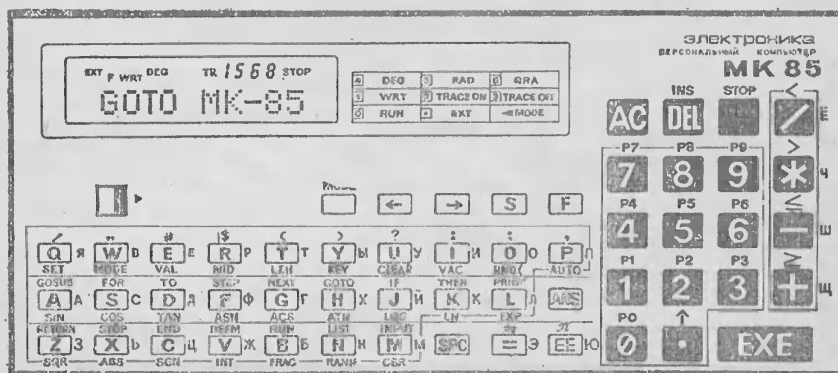


Рис. 2. Расположение информации на индикаторе ПК «Электроника МК 85»

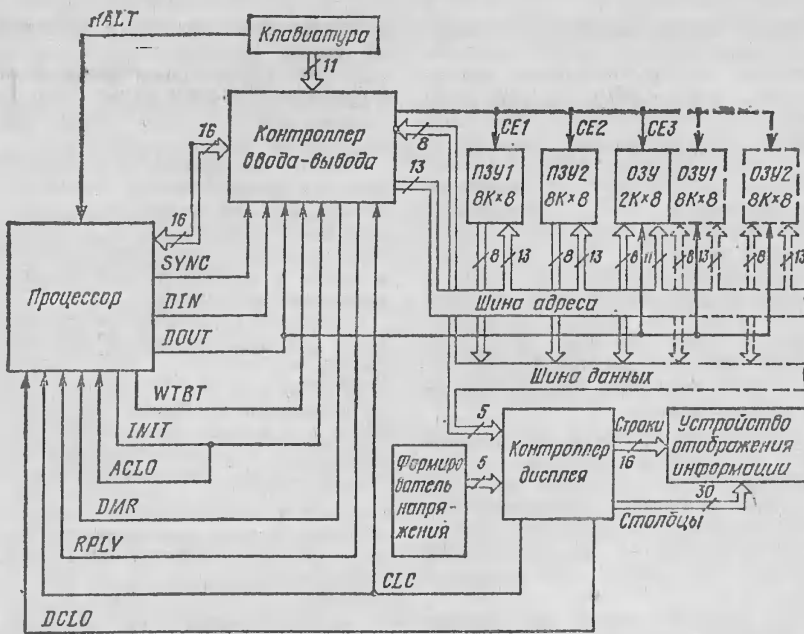


Рис. 3. Функциональная схема ПК «Электроника МК 85»

Кроме клавиатуры и индикатора на верхней панели расположен выключатель питания, а в левом торце — разъем для подключения блока питания.

Компьютер реализован на основе однокристалльного микропроцессора и комплекта интерфейсных КМДП БИС (рис. 3).

Процессор представляет собой однокристалльный 16-разрядный микропроцессор, изготовленный по КМОП-технологии и предназначенный для обработки цифровой информации (рис. 4). Система счисления для чисел и команд — двоичная. Разряд-

ность для чисел и команд — 16 двоичных разрядов. Система команд — безадресная, одноадресная, двухадресная. Виды адресации — регистровая, косвенно-регистровая, автоинкрементная, косвенно-автоинкрементная, индексная, косвенно-индексная. Число регистров общего назначения — 8. Число каналов передачи информации — 1. Число команд — 72. Объем адресуемой памяти — 64К байт. Максимальная тактовая частота — 2 МГц.

Шина адрес-данные представляет собой 16 входов и выходов системной магистрали, совмещенной по адресам и данным. Использование

одних и тех же выводов для передачи адресов и данных достигается путем разделения во времени. Передача «Jog.1» соответствует низкий уровень сигнала на этих выводах.

Синхронизация обмена данными между процессором и запоминающими устройствами осуществляется при помощи канальных сигналов SYNC, DIN, DOUT, WTBT, RPLY, DMR — вход сигнала запроса на прямой доступ к памяти. ACLO, DCLO — входы, используемые для начального запуска процессора. HALT — вход прерывания выполнения программы.

Контроллер ввода-вывода (рис. 5) изготовлен по КМОП-технологии и выполняет следующие функции: обмен данными между процессором и накопителями ОЗУ и ПЗУ, регенерацию данных для формирования информации на ЖКИ, ввод данных с клавиатуры, приостанов процессора во время ожидания ввода для экономии энергии элементов питания.

Адресное пространство компьютера с размером 18К байт распределяется следующим образом: ПЗУ1 — 0...17777, ПЗУ2 — 20000...37777, ОЗУ — 40000...43777.

Область ОЗУ 40000...40137 предназначена для хранения изображения, формируемого на индикаторе (экранное ОЗУ). По адресам 40140...41471 располагается системная область. ОЗУ пользователя, расположенное по адресам 51472...43777, позволяет хранить программы длиной 1221 шаг. Для увеличения объема хранимых программ предусмотрена возможность расширения ОЗУ пользователя до 47777 (7365 шагов) и до 77777 (15557 шагов) при использовании соответственно одной или двух микросхем ОЗУ с организацией 8Кx8 (см. рис. 3).

Обмен данными между контроллером и памятью происходит по 8-разрядной шине данных. На один обмен 16-разрядными данными по системному каналу процессора приходится два обмена 8-разрядными данными по шине данных ОЗУ или ПЗУ. Адресация обменов осуществляется по отдельной шине адреса и синхронизируется сигналами управления CEROM1, CEROM2, CERAM1, CERAM2.

Для ввода в процессор данных с клавиатуры в контроллер встроено 11-разрядный регистр, доступный по чтению. Информация в этом регистре хранится все время нажатия любой клавиши.

Через каждые 0,4 мс контроллер низким уровнем сигнала DMR тормозит выполнение программы процессора, требуя прямого доступа к памяти. Автоматически на шине адреса контроллер формирует серию адресов определенной частоты ОЗУ, сигнал CERAM1 и сигнал CENG (синхронизирующий данные, форми-

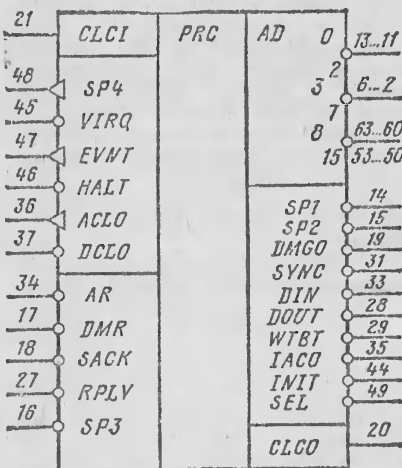


Рис. 4. Условное графическое изображение микропроцессора T243-2

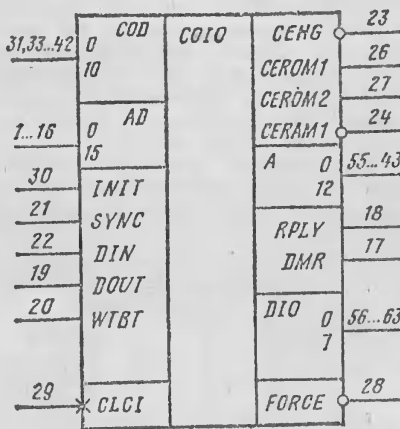


Рис. 5. Условное графическое изображение контроллера ввода-вывода T241-2-015

рующее изображение на ЖКИ) для каждого адреса. Таким образом, контроллер является управляющим устройством в процедуре регенерации изображения на индикаторе. Контроллер дисплея (рис. 6) изготовлен по КМОП-технологии и предназначен для управления ЖК-индикатором матричного типа с организацией 30×16 элементов разложения, работающего в мультиплексном режиме, генерации тактового сигнала с частотами 250, 500 кГц и 2 МГц, формирования сигнала запуска процессора DCLO.

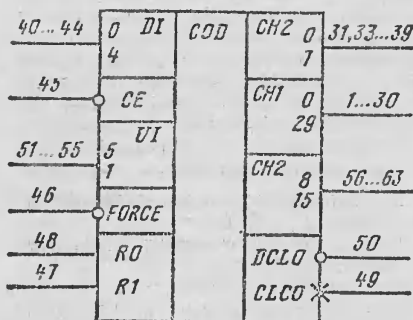


Рис. 6. Условное графическое изображение контроллера дисплея T241-2-014

В зависимости от данных, входящих на входной регистр, микросхема T241-2-014 формирует на выходе комбинацию сигналов, которые позволяют получать на ЖК-индикаторе информацию о входных сигналах.

В начальный момент работы все узлы микросхемы переводятся в исходное состояние низким уровнем напряжения на выводе 51. С появлением высокого уровня на этом выводе БИС переходит в рабочий режим, включается тактовый генератор, через 2048 тактов схемой RDCLO формируется высокий уровень напряжения на выводе DCLO, снимается высокий уровень с входов R всех триггеров схемы.

При появлении низкого уровня сигнала CE на входе БИС во входном регистре ID захлопываются данные с входов D0...D4. По срезу импульса CE (высокий уровень) в одной из шести частей CODU захлопываются данные, записанные перед этим в ID. Этот процесс происходит поочередно со всеми частями CODU, т. е. с приходом первого сигнала сопровождения данных CE выбирается первая часть, с приходом второго — вторая и т. д. Имеющийся в CODU счетчик до шести наполняется фронтом сигнала CE так, что по седьмому сигналу данные будут записаны в первую часть и т. д.

После записи данных в последнюю, шестую, часть CODU вырабатывается сигнал управления коммутатором строк CH и формируется сигнал выбора первой строки. Каждая следующая строка выбирается после очередного заполнения CODU. Счетчик до 16 в узле CH, аналогично счетчику в ODU, наполняется фронтом сигнала управления коммутатором строк CH так, что после шестнадцатой строки будет выбрана первая строка. Кроме того, по фронту 16-го сигнала управления коммутатором строк CH узел управления фазой выходных напряжений CF меняет фазу выходных напряжений на противоположную.

Коммутатор выходных напряжений COU работает по сигналам, поступающим с узлов CF, CH и ODU, а также использует аналоговые сигналы (1...5 В) для формирования импульсов, служащих для непосредственного управления ЖК-индикатором.

При высоком уровне на выводе 46 FORCE тактовый генератор формирует на выводе 49 CLCO меандр частотой 250 или 500 кГц в зависимости от значения данных, сопровождаемых каждым шестнадцатым сигналом CE, начиная с первого. В этом случае, если D4=0 (низкий уровень), формируется меандр 250 кГц, если D4=1 — меандр 500 кГц. Переключение с одной частоты на другую происходит без проколов и выбросов на выводе CLCO.

При низком уровне на выводе 46 FORCE тактовый генератор формирует на выводе 49 CLCO меандр частотой 250 кГц или 2 МГц в зависимости от значения данных, сопровождаемых каждым шестнадцатым сигналом CE, начиная с первого. При переключениях состояния вывода FORCE возможно появление кратковременных изменений скважности на выводе CLCO.

ПЗУ представляет собой две микросхемы T242-2, изготовленные по КМОП-технологии с организацией 8K×8 (рис. 7), ОЗУ — КМОП-микросхема T244-2 статического типа с организацией 2K×8 (рис. 8).

Конструктивно компьютер выполнен на одной двухсторонней печатной плате. Цепи ввода информации

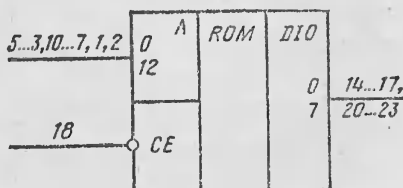


Рис. 7. Условное графическое изображение микросхемы ПЗУ T242-2

коммутируются контактными парами на основе токопроводящего эластомера, что обеспечивает высокие конструктивно-технологические характеристики компьютера и гарантирует надежность при эксплуатации.

Память компьютера в отличие от обычных калькуляторов энергонезависима, т. е. ее содержание сохраняется неопределенно долгое время при установленных элементах питания, а также в течение 15 мин после их извлечения для замсны.

Невысокая потребляемая мощность (0,02 Вт) обеспечивает непрерывную работу «Электроники МК 85» от элементов питания в режимах записи и отладки программ в течение 200 ч, а в режиме вычислений — в течение 80 ч. Предусмотренный в конструкции режим ускоренного в 4 ра-

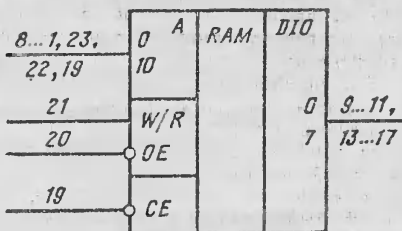


Рис. 8. Условное графическое изображение микросхемы ОЗУ T244-2

за быстродействия значительно увеличивает потребляемую мощность, поэтому его рекомендуется использовать только при внешнем питании.

На ПК «Электроника МК 85» установлен гарантийный срок 24 месяца. Его розничная цена 135 руб. Прилагается к компьютеру руководство по эксплуатации включает в себя основы программирования на используемой версии языка Бейсик. Методика программирования изложена в следующей статье данного раздела. Кроме того, изготовитель готовит к выпуску библиотеки программ на этом языке в целом ряде возможных сфер использования микрокомпьютера «Электроника МК 85».

Телефон для справок: 532-83-31, Москва

Статья поступила 7 апреля 1987 г.

ПРОГРАММИРОВАНИЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА «ЭЛЕКТРОНИКА МК 85» С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА БЕЙСИК

«Электроника МК 85» является первым отечественным карманным персональным компьютером, использующим для программирования широко распространенный язык Бейсик.

Версия Бейсика для ПК «Электроника МК 85» учитывает, что он работает без внешних устройств ввода-вывода и хранения информации. Кроме того, на встроенный жидкокристаллический дисплей можно выводить одновременно до 12 различных информационных символов, а в режиме бегущей строки 63 символа (не считая служебной информации о режимах работы ЭВМ). Это накладывает определенные ограничения на использование ряда известных в Бейсике базовых и системных операторов (команд). Вместе с тем в версию Бейсика для МК 85 введены некоторые команды, не используемые в других версиях языка, а ряд операторов имеет несколько иное по сравнению с общепринятым значение.

Рассмотрим некоторые особенности и основные возможности программирования на компьютере МК 85, сочетающем свойства обычного калькулятора и программируемой микроЭВМ. В режиме ручных вычислений (прямой режим) с его помощью можно выполнять любые арифметические операции, введенные с клавиатуры в обычной математической форме с контролем на дисплее, в порядке, определяемом их приоритетами. Сначала выполняются алгебраические, тригонометрические и другие функции, затем возведение в степень, умножение и деление и последними — сложение и вычитание. Выражения, записанные в скобках, получают высший приоритет. При одинаковом приоритете вычисления производятся слева направо.

Компьютер «Электроника МК 85» имеет следующие встроенные функции, выполняемые как в прямом, так и в программном режимах: тригонометрические и обратные тригонометрические функции с заданием аргумента в градусах, радианах или градах (1 градус = 0,9 градуса); логарифмические функции; вычисление квадратного корня и степени числа; округление с заданной точностью, определение целой и дробной частей числа; вычисление модуля числа; перевод чисел из десятичной формы в экспоненциальную; генерирование случайных чисел, равномерно распределенных в интервале от 0 до 1; число π ; число e .

При реализации перечисленных функций диапазоны изменения аргумента имеют значения, указанные в таблице. Точность вычислений составляет ± 1 в десятом знаке.

Функция	Диапазон изменения аргумента
$\sin X$, $\cos X$, $\operatorname{tg} X$ $\arcsin X$, $\arccos X$ $\operatorname{arctg} X$ $\log X$, $\ln X$ e^X \sqrt{X} X^Y ($X \uparrow Y$)	$ X < 1440^\circ$ (90° рад; 1600 град) $ X < 1$ $ X < 1 \cdot 10^{10}$ $0 < X < 1 \cdot 10^{10}$ $-9429 < X < 9429$ $0 < X < 1 \cdot 10^{10}$ $ X < 1 \cdot 10^{10}$ при $X < 0$, Y — целое число

Компьютер работает в восьми возможных режимах, коды которых указаны на передней панели:

- RUN — режим калькулятора или выполнения записанной в память программы;
 WRT — режим записи программы в память, ее проверки и редактирования;
 TRACE ON — ввод в режим построчной отладки программы;
 TRACE OFF — вывод из режима построчной отладки программы;
 DEG — задание угловых величин в градусах;
 RAD — задание угловых величин в радианах;
 GRA — задание угловых величин в градах;
 EXT — ввод или вывод из режима расширения функциональных возможностей (использование для комментариев прописных или строчных букв русского алфавита, а также специальных символов, изображенных на наклейке клавиатуры).

Кроме того, предусмотрены четыре регистра для изменения вида работы с клавиатурой:

а) обычный — с клавиш вводятся символы, написанные непосредственно на них;

б) верхний (Shift) — вводятся операторы, символы или номера программных зон-файлов (P0...P9), написанные над клавишами. Для работы с этим регистром служит функция S;

в) нижний (Function) — вводятся функции, написанные под клавишами. Для работы с этим регистром служит функция F;

г) режимный (MODE) — вводятся буквы русского алфавита, написанные справа от клавиш, или специальные символы, изображенные на наклейке клавиатуры.

Операторы и функции, для которых имеются специальные клавиши на клавиатуре МК 85, можно вводить и побуквенно, как на других ЭВМ. Для примера рассмотрим работу с клавиатурой и дисплеем МК 85 при вычислении некоторых функций в режиме калькулятора.

Пример 1. Вычислить выражение

$$12 + 36 - \frac{9 \times 5}{4}$$

Задан прямой режим нажатием клавиш MODE и 0. На дисплее индицируется надпись RUN. Введем требуемое выражение клавишами

1 2 + 3 6 - 9 * 5 / 4

контролируя правильность ввода по дисплею. Наждем клавишу выполнения операции EXE и на дисплее увидим результат 36,75.

Пример 2. Перемножить числа 12,3 и 4,56 и результат округлить до одного знака после запятой, отбросив вторые и следующие знаки.

Вводим с клавиатуры код прямого режима, команду округления RND (по буквам или с помощью специальной клавиши в нижнем регистре) и формулу перемножения двух чисел;

MODE 0 RND (1 2 3 * 4 . 5 6) RND EXE

На дисплее читаем ответ 56,1.

Пример 3. Найти дробную часть результата деления 4550 на 16.

Используем команду выделения дробной части числа FRAC. Вводим с клавиатуры:

MODE 0 [F] FRAC 5 (4 5 5 0 / 1 6 5) EXE

Ответ на дисплее 0.375.

При работе в программируемом режиме в память МК 85 можно ввести 1...10 программ общим объемом до 1221 шагов и постоянно хранить, так как память энергонезависимая. Число шагов, занимаемых программой, подсчитывается по следующему правилу: программная команда соответствует одному шагу на команду; функциональная команда — одному шагу на команду; № строки программы — двум шагам на строку; символ (буква, цифра, знак) — одному шагу на символ; нажатие клавиши EXE — аналог привычной «возврат каретки» — одному шагу.

Например, строка вида

$\frac{300}{2}$ FOR I = 1 TO N

занимает 8 шагов в памяти. Максимальная длина строки 63 символа позволяет записывать в ряде случаев в нее несколько операторов, экономя память ЭВМ. В режиме записи программы WRT для облегчения контроля на дисплее отображается число шагов, не использованных в памяти.

Обычный режим работы позволяет использовать в программе до 26 переменных (число букв латинского алфавита на клавиатуре). Но в режиме расширения числа переменных по оператору DEFM появляется возможность работы программы с одномерными массивами индексированных переменных, максимальное число которых, занимающих всю память ЭВМ, равно 178. Каждая вновь вводимая (сверх 26) переменная уменьшает объем памяти на 8 шагов.

МК 85 может работать и с текстовыми переменными (строками) длиной до семи символов (букв, цифр, знаков), для чего используется знак \square (аналог знака & в других вариантах Бейсика).

Всего в компьютере используются 21 оператор: INPUT, PRINT, GOTO, FOR, NEXT, IF-THEN, GOSUB, RETURN, STOP, END, RUN, LIST, MODE, SET, VAC, CLEAR, CLEARA, DEFM, AUTO, DRAW, DRAWC и 9 специальных программных функций: KEY, CSR, LEN, MID, VAL.

1. Операторы ввода. Оператор INPUT. Вид: INPUT <«обозначение переменной», переменная> или INPUT <переменная>. Например:

20 INPUT «число», X, «N», N, Y EXE

Предназначен для запроса значения переменных, вводимых с клавиатуры во время выполнения программы. При этом работа программы приостанавливается и на дисплее появляются названия требуемых переменных с вопросительным знаком или просто вопросительный знак, если в операторе не записано «обозначение переменной». Ввод значений каждой переменной осуществляется через EXE. После последнего нажатия на EXE работа программы продолжается.

Оператор KEY. Вид: <текстовая переменная> = KEY. Например:

10A \square = KEY : IF A \square = «X» THEN 30 EXE

Предназначен для ввода с клавиатуры значений текстовой переменной, соответствующей клавишам, нажатым во время отработки оператора.

2. Операторы вывода. Оператор PRINT. Вид: PRINT <«текст или обозначение переменной», переменная>. Например:

100 PRINT «длина»; X; «Y»; Y EXE

Предназначен для вывода на дисплей вычисленных значений и комментария к ним латинскими и русскими прописными или строчными буквами.

Оператор CSR. Используется в операторе PRINT и

предназначен для задания требуемого формата вывода информации на дисплей. Определяет величину промежутка на дисплее между выводимыми числами или символами. Записывается в виде:

100 PRINT «длина»; X; CSR N; «Y»; Y EXE

В этом случае на дисплее между первой цифрой в значении X и буквой «Y» будет сделан пробел ровно в N мест.

3. Операторы условных и безусловных переходов. Оператор GOTO. Вид: GOTO <N строки>. Служит для безусловной передачи управления на строку программы, номер которой записан в этом операторе. Причем номер N строки может быть дан не только числом, но и в виде выражения, значение которого должно быть вычислено в ходе выполнения программы. Например:

70 GOTO A*B+40 EXE

Оператор IF...THEN. Вид: IF <условие> THEN <N строки>. Предназначен для передачи управления на строку с номером N при выполнении условия, записанного в операторе. Например:

30 IFA \geq B + C THEN 80 EXE

При невыполнении этого условия управление передается на следующий по порядку оператор. Вместо номера строки после THEN может стоять # n (n=0...9). Это значит, что при выполнении условия управление передается другой программе, записанной в файле Pn (условный переход к подпрограмме). Оператор IF может использоваться и в виде IF <условие>; <оператор, команда, математическое выражение>. Тогда при выполнении условия начинает выполняться заданный оператор или команда, либо вычисляется записанное математическое выражение.

Оператор GOSUB. Вид: GOSUB <N строки>. Используется для безусловного перехода к подпрограмме, первая строка которой имеет номер N в том же файле, что и основная программа. Вместо номера первой строки подпрограммы может быть записан номер другого файла, в который помещена подпрограмма (как в операторе IF при условном переходе к подпрограмме). Например:

60 GOSUB (<# Pn>) EXE

Оператор RETURN. Оператор возврата от подпрограммы к основной программе. Обязательно записывается в конце подпрограммы и обеспечивает возврат управления к оператору, следующему за оператором GOSUB (по которому была вызвана данная подпрограмма).

4. Операторы циклов. Оператор FOR и NEXT. Вид: FOR <переменная>-n TO m STEP l [...] NEXT <переменная>, где n, m, l — некоторые численные значения данной переменной. Используется для обозначения начальной и конечной точек цикла программы, т. е. в тех случаях, когда необходимо повторить выполнение некоторых операторов, обозначенных выше квадратными скобками, для значений переменной, изменяющихся от n до m с шагом l. Как только значение переменной превысит по величине m, управление передается оператору, следующему за NEXT. Например:

20 FOR X=1 TO 20 STEP 2

[30 ...]
[:]
[60 ...]

70 NEXT X

Если шаг l=1, то его можно не записывать, т. е.

20 FOR X = 1 TO 20

[:]

70 NEXT X

Бейсик для МК 85 предусматривает возможность использовать в одной программе до четырех циклов, vlo-

женных друг в друга. Нельзя строить циклы, пересекающие друг друга, например:

```
FOR X = 1 TO 19 STEP 2
  FOR Y = 2 TO 20 STEP 3
    ...
  NEXT X
  ...
NEXT Y
```

Для выхода из цикла используют операторы условного перехода IF THEN.

5. Оператор остановки вычислений STOP. Предназначен для остановки вычислений в нужном месте программы с целью запроса с клавиатуры каких-то промежуточных вычислений значений. При наличии в программе цикла оператор STOP после нужной строки останавливает работу ЭВМ. Затем при наборе с клавиатуры названия вычисленной в цикле переменной выдает ее текущее значение на дисплей. Например:

```
20 FOR X = 1 TO 50 STEP 2
30 Y = X ↑ 2 + 15
35 STOP
40 NEXT X
```

При этом на каждом шаге цикла после остановки нужно нажать клавишу Y, чтобы увидеть его текущее значение. Оператор STOP обычно используется для отладки и контроля выполнения программы.

6. Оператор окончания программы END. Используется для обозначения конца выполнения программы и записывается в последней строке. Служит чаще всего признаком окончания конца программы и может быть опущен, кроме случаев обращения к другим программам и подпрограммам с помощью операторов GOTO и GOSUB.

7. Оператор обнуления VAC. Предназначен для предварительного обнуления всех информационных ячеек памяти, если в них при работе программы записываются результаты вычислений.

8. Оператор вызова программы на дисплей LIST. Записывается в виде LIST (N строки). При этом на дисплей выводится программа (листинг), начиная со строки с заданным номером. Если оператор LIST используется в режиме RUN, то программе можно только просмотреть при автоматическом «пролистывании» всех строк в режиме бегущей строки. Если в режиме WRT, то ее можно и редактировать, так как переход к просмотру новой строки происходит только после нажатия на EXE.

9. Оператор выполнения программы RUN. Записывается в виде RUN (N строки) и сам не нумеруется. Предназначен для запуска выполнения программы с начала (без (N строки)) или с нужной строки после ее окончания.

10. Операторы стирания. Оператор CLEAR. Предназначен для стирания программы, записанной в том файле Pn, с которым работает в данный момент МК 85.

Оператор CLEARA. Предназначен для стирания программ во всех файлах P0...P9.

Эти операторы выполняют ту же функцию, что и оператор DELETE (исключение из архива) в других версиях Бейсика. В МК 85 тоже есть клавиша DEL, но она позволяет стирать только один символ при редактировании программы со сдвигом остального содержимого строки на одно место влево.

11. Оператор режима MODE. Вид: MODE (код режима). Список режимов работы МК 85 приведен ранее.

12. Оператор задания формата числа SET. Вид: SET (p), где $0 \leq p \leq 9$. Означает ограничение числа знача-

щих цифр в результате числом p. Для отмены этого указания необходимо ввести в программу оператор SET 10.

13. Операторы преобразования текстовых переменных (отсутствуют в ряде других вариантов Бейсика).

Оператор LEN. Записывается в виде (переменная) = LEN («текстовая переменная»). Предназначен для определения числа символов в значении текстовой переменной. Для того чтобы вывести на дисплей эту величину, после оператора LEN необходимо записать оператор PRINT (переменная).

Оператор MID. Записывается в виде MID ((m, n)), где m и n — целые числа 1...30 или математические выражения. Предназначен для выделения из строки символов, обозначающих текстовую переменную, p символов, начиная с m-го. Например, если \$ = «ABCDEFGHIJ», то операторы

```
50 X$ = MID (3, 4)
```

```
60 PRINT X$
```

выделят из нее и выведут на дисплей символы CDEF. Если в операторе MID не задать величину n, то программа выделит из \$ все символы, начиная с m-го.

Оператор VAL. Записывается в виде VAL («текстовая переменная»). Служит для замены в программе текстовой переменной, состоящей из цифр и десятичной точки, на ее численное значение. Например, если Z = «35482», то VAL (Z) = 35482.

14. Оператор автоматической нумерации строк программы AUTO. Используется для автоматической последовательной нумерации строк программы с заданным шагом. Записывается в виде AUTO (шаг нумерации строк).

При работе с МК 85 имеются широкие возможности редактирования и исправления ошибок в программе как с помощью операторов Бейсик, так и непосредственно с клавиатуры. Для этого используются клавиши передвижения курсора →, ←, клавиша очистки дисплея AC, стирания лишнего или введение пропущенного символа INS DEL.

В компьютере имеется также развитая система диагностики, распознающая и индицирующая на дисплее семь видов ошибок, связанных с переполнением памяти, ошибками синтаксическими, математическими, ошибками в построении циклов, задании переменных и аргументов, обращении к строкам программы и подпрограммам. Причем МК 85 извещает пользователя о типе ошибки, номере строки и файла, в которых имеется ошибка.

Кроме того, на индикаторе в процессе записи и выполнения программы отображается разнообразная сервисная информация — обрабатываемый режим, регистр клавиатуры, номер файла, к которому обращается пользователь, свободен ли оп, количество использованных шагов программы.

Все это значительно облегчает работу с «Электроникой МК 85» любому пользователю, как знакомому с другими версиями Бейсика, так и новичку в программировании на этом языке.

Телефон для справок: 158-43-67, Москва.

ЛИТЕРАТУРА

- Кетков Ю. Л. Программирование на Бейсике. — М.: Статистика, 1978.
Уорт Т. Программирование на языке Бейсик. — М.: Машиностроение, 1981.
Куликов В. Д. Курс программирования. — Ленинград: ЛГУ, 1982.

Статья поступила 15 декабря 1986 г.

УДК 681.012.011.56

С. В. Казменко

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ СИСТЕМ НА ОСНОВЕ ОБЪЕДИНЕНИЯ КОНЦЕПЦИИ МАКЕТИРОВАНИЯ И «ПРОГРАММИРОВАНИЯ БЕЗ ПРОГРАММИСТА»

Одним из основных качественных изменений в сфере автоматизированной переработки информации является в будущем возможность для конечных пользователей создавать персонально-ориентированные системы, извлекать информацию из баз данных и работать с этой информацией, не обращаясь к посредникам [1]. В настоящей статье рассматривается подход, позволяющий осуществлять реализацию этой возможности (обойтись без посредника) для весьма широкого класса приложений. **Общая постановка задачи**

Все многообразие способов создания прикладных систем (или программ), автоматизирующих конкретные задачи конечного пользователя, разобьем на два класса: способы, которые так или иначе могут быть отнесены к программированию; способы настройки готового программного обеспечения. Например, подготовка командного файла, состоящего только из вызовов готовых программ, может рассматриваться как создание линейной программы на языке, содержащем операторы единственного типа (вызов подпрограммы).

Определим «программирование без программистов» [8] как деятельность, позволяющую получить требующийся программный продукт, не прибегая к составлению программы как определенной совокупности связанных между собой операторов. Таким образом, в статье «программирование без программиста» фактически отождествляется с настройкой, точнее, с ее наиболее развитыми формами.

Как отмечается в [2], «реальные возможности профессиональных программистов в ближайшем будущем будут ограничиваться разработкой базовых средств и лишь наиболее универсальных пакетов программ... Все, что могут сделать профессиональные программисты..., это... разработать инструментальные средства, облегчающие непрограммирующим профессионалам процесс самостоятельной формализации их индивидуальных знаний». Упомянутые в [2] инструментальные средства позволяют осуществлять настройку базового програм-

мно обеспечения. Поэтому будем считать, что каждая конкретная система, предоставляющая возможность автоформализации профессиональных знаний, реализует для некоторого класса задач технологию «программирования без программиста».

Концепция макета программы (программной системы) появилась в начале 80-х годов [2] как альтернатива традиционному подходу к проектированию, для которого типичной была следующая последовательность событий: конечным пользователем заказывалась программная система для решения конкретной задачи; создавалась эта система; в начале эксплуатации системы выявлялись настолько значительные и многочисленные разночтения между заказчиком и программистом в постановке задачи, что стоимость внесения необходимых изменений была сопоставима с повторной разработкой системы.

Как правило, каждое из выявленных разночтений было сравнительно легко (по сравнению с разработкой системы заново) устранено. Это обстоятельство в совокупности с тем, что разночтения в задании на программирование обычно выявлялись постепенно, побуждало осуществлять многочисленные изменения программной системы. В результате в трясину сопровождения засасывало большое число квалифицированных программистов, а окончательная стоимость разработки во много раз превышала первоначальную.

Практически только при работе с готовой системой пользователь начинает активно формулировать понятные программисту требования к ней, кроме того, работа с системой обычно приводит пользователя к новому пониманию своих задач и, в свою очередь, к изменению требований.

Изнуряющий традиционный подход к созданию программного обеспечения в совокупности с необходимостью как можно скорее получить от пользователя вразумительный ответ о том, какая система ему нужна, привели к появлению концепции макета, т. е. такой программы, единственное назначение которой — демонстрация пользователю возможностей будущей

системы для выработки взаимопонимания между пользователем и программистом.

В данной статье предлагается осуществлять макетную настройку базовых программных средств для демонстрации пользователю системы, полученной в результате этой настройки. В работе [3] такой макет называется демонстрационной моделью. Этим термином и будем пользоваться, чтобы подчеркнуть отличие демонстрационной модели от обычной программы-макета.

Традиционный подход к созданию прикладных систем характеризуется наличием трех основных действующих лиц: конечного пользователя (заказчик программного обеспечения), постановщика задачи (системный аналитик) и программиста. Идеальной следует считать такую ситуацию, когда пользователь сам является постановщиком, владеет возможностями настройки и сам создает первый макет. Если при этом имеется возможность получить какой-то вариант работающей системы, не пользуясь средствами традиционного программирования, то по существу это можно считать одним из способов воплощения концепции автоформализации профессиональных знаний [2]. В случае, когда пользователь не обладает необходимыми для этого навыками, в роли постановщика выступает системный аналитик, причем такую ситуацию можно рассматривать в качестве переходного этапа, на котором пользователь сам овладевает навыками настройки.

Автоматизация разработки экранных документов

Рассмотрим применение предложенного подхода на примере системы автоматизации проектирования входных экранных документов. При этом конкретный экранный документ однозначно задается двумерной экранной картинкой и таблицей описания реквизитов*.

* В простейшем случае, когда для каждого реквизита задается всего два параметра (длина и экранный адрес), описание реквизитов может быть составной частью экранной картинки (например, реквизиты могут выделяться на экране с помощью скобок []). При этом возможно использование обычного экранного редактора в качестве средства проектирования таких документов. В более сложных случаях, когда каждый реквизит определяется большим числом параметров, их «втискивание» в экранный бланк становится неудобным и возникает потребность в разделении настраиваемой информации на бланк документа и таблицу описания реквизитов. Рассматриваемая система позволяет определять несколько большее число параметров, чем описывается в настоящей статье.

Интерактивный генератор программ форматного ввода позволяет пользователю-непрограммисту выполнить следующие действия [4]:

разметить экран (сформировать бланк входного экранного документа). Пользователь просто «рисует» этот бланк, используя все символы клавиатуры дисплея VT 340;

сформировать таблицу реквизитов. Параметры реквизитов вписываются пользователем в эту таблицу, представленную на экране. Заполнение некоторых параметров автоматизировано, например, экранный адрес реквизита можно сформировать, выбрав курсором местоположение первого символа на готовом бланке экранного документа. После этого на экране восстанавливается таблица, в которую автоматически вносятся координаты начала реквизита на экранной форме;

автоматически построить часть таблицы. При этом некоторые параметры заполняются стандартными значениями;

оттестировать правильность сформированной управляющей информации (бланка экранной формы и таблицы реквизитов) — разработчику предоставляется возможность поработать с новым экранным документом, т. е. пользователь сам себе демонстрирует макет;

сформировать готовую управляющую информацию в виде загрузочного модуля, т. е. получить работающую программу;

настроиться на ранее подготовленную с помощью генератора программу для изменения бланка или каких-то значений в таблице реквизитов.

Дальше рассматривается упрощенная по сравнению с реально существующей версия системы [4]. В начале работы с системой на экране появляется следующее меню:

формирование экрана и таблицы, тестирование документа, получение готовой программы, прием программ для исправления. Разработчик выбирает ту функцию, которую собирается выполнить. После выполнения каждой функции происходит возврат в исходное состояние, а на экране восстанавливается меню.

Последовательность выполнения функций не регламентирована, в результате чего имеется возможность итерационной разработки экранного документа.

Пусть необходимо разработать экранный документ для ввода информации учета программного обеспечения. Для этого решено, что по каждому модулю (программе) будут вводиться следующие показатели: руководитель разработки (фамилия и инициалы); подразделение, в котором он работает (шифр); программист (фамилия и инициалы); его подразделение (шифр); имя модуля (программы); тип ЭВМ; объем требуемой оперативной памяти,

Вызвав функцию формирования экрана, проектировщик может сформировать («нарисовать», используя все символы клавиатуры) бланк (рис. 1).

Руководитель:	/
Программист:	/
Имя модуля (программы):	
ЭВМ:	Объем оп.пам.: К байт

Рис. 1. Разметка экрана (8×36)

Сформировав бланк входного документа, проектировщик возвращается в исходное состояние (меню) и вызывает вторую функцию (формирование таблицы). Перед началом перехода к ее заполнению система при необходимости (в случае первого вызова второй функции) запрашивает у проектировщика общее число показателей во входном документе. После этого на экране появляется таблица (рис. 2), в которую проектировщик должен вписать параметры показателей проектируемого экранного документа.

Заполните таблицу показателей	№	Имя реквизита	Экранный адрес
	1	16	013
	2	4	032
	3	16	113
	4	4	112
	5	10	222
	6	6	305
	7	3	329

Рис. 2. Заполнение таблицы показателями входного документа

Сформировав бланк экранного документа и таблицу его показателей, проектировщик может вызвать функцию тестирования документа. В спроектированном документе (рис. 3) определены все входящие в него информационные показатели в отличие от бланка (рис. 1).

Проектировщику при тестировании предоставляется возможность взаимодействия с демонстрационной моделью, которая может либо полностью удовлетворить его, либо побудить к доработке этой модели. На-

Руководитель:	Иванов /
Программист:	Петров /
Имя модуля (программы):	
ЭВМ:	Объем оп.пам.: К байт

Рис. 3. Спроектированный входной экранный документ

пример, проектировщик может при тестировании выявить ошибку проектирования или несоответствие реальных данных, которые будут вводиться в документ (см. рис. 2), заданным параметрам, что потребует внесения исправлений в модель. Если в качестве проектировщика выступал постановщик, то, завершив разработку модели, он демонстрирует ее заказчику.

На примере рассмотренной системы проиллюстрируем еще одно важное качество предлагаемого подхода — это возможность моделирования в явном виде. С помощью инструментальной системы [4] без участия программиста на ЭВМ ЕС-1010 получается готовое программное обеспечение для микроЭВМ VTS-56100 [5]. Однако эту систему можно использовать не только по своему прямому назначению. Если у конечного пользователя установлены другие технические средства, для которых с помощью рассмотренной инструментальной системы не может быть получено программное обеспечение, то возможен следующий путь:

с помощью инструментальных средств проектируется демонстрационная модель; заказчик знакомится с ней и подтверждает свою потребность именно в таких, идентичных модели, программных средствах;

программистом реализуется запрошенное заказчиком (и согласованное с ним на демонстрационной модели) программное обеспечение для его технических средств.

Автоматизация разработки средств персональных вычислений

Функциональные возможности рассмотренной инструментальной системы позволяют проектировать лишь часть программного обеспечения, необходимого конечным пользователям. Более широкое применение предлагаемого подхода возможно на примере системы персональных вычислений (СПВ), которая реализована в виде программы-макета.

Система персональных вычислений содержит в качестве своей составляющей диалоговую подсистему ведения информационной базы. Эта подсистема позволяет:

формировать массив записей (транзакций), каждая из которых представляется пользователю в виде экранного документа (рис. 4), аналогичного проектируемому с помощью рассмотренной инструментальной системы;

осуществлять поиск нужных записей, т. е. пользователь должен сформировать поисковый образец, для чего он нажимает условную клавишу ? после этого на экране появляется точно такой же бланк (см. рис. 4), как при заполнении (просмотре) записей.

В окна бланка вписаны параметры поискового образца, в качестве кото-

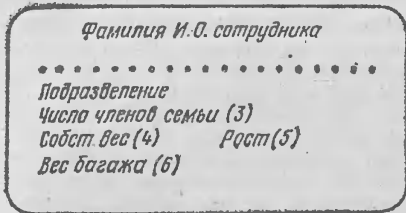


Рис. 4. Представление записи на экране

рых выступают значащие (отличные от пробелов) алфавитно-цифровые символы. Если пользователь хочет выбрать из информационного массива только записи, содержащие фамилии, начинающиеся с буквы К, то он должен ввести в поисковый образец эту единственную букву в первую позицию реквизита ФАМИЛИЯ. После заполнения образца осуществляется поиск: нажимается клавиша «;» и система последовательно просматривает записи массива. При обнаружении записи с фамилией на букву К поиск прекратится и на экране будет представлена найденная запись. При повторном нажатии «;» поиск продолжится до следующей записи. Фактически такой механизм поиска по образцу является сильно упрощенным вариантом языка запросов QUERY-BY-EXAMPLE [6].

Персональные вычисления в СПВ производятся после перехода в режим диалоговых вычислений. При этом на экране в правой части верхней строки появляется сообщение ПЕРСОНАЛЬНЫЕ ВЫЧИСЛЕНИЯ, а в нижней строке экрана — содержимое шести числовых регистров ($R_i, i=0,5$).

В режиме диалоговых вычислений пользователю доступны следующие команды:

преобразование одного из реквизитов записи в число и его размещение (засылка) в R_0 ;

пересылка числа из R_0 в R_1 ($i=1,5$);

пересылка числа из R_1 в R_0 ($i=1,5$);

сложение содержимого R_0 с содержимым R_1 ($i=1,5$) и засылка результатов в R_0 ;

диалоговое формирование числа и его засылка в R_0 ;

выход из режима персональных вычислений.

Большинство этих команд задается посредством нажатия на две клавиши. Так, для задания команды преобразования реквизита в число (с помещением в числовой регистр) нужно сначала нажать клавишу ↓, затем цифру, указывающую на конкретный реквизит записи. Команды пересылки и сложения задаются соответственно клавишами →, ←, +, после которых необходимо задать цифру 1..5 для идентификации второго регистра (первым является R_0),

участвующего в операции. При инициализации одной из команд ↓, →, ←, + перед нажатием цифры в левом верхнем углу экрана появляется одно из следующих сообщений: ВЫБЕРИТЕ РЕКВИЗИТ, $R_0 \rightarrow R(?)$, $R(?) \rightarrow R_0$, $R(?) + R_0 \rightarrow R_0$. Эти сообщения позволяют визуально идентифицировать незавершенную текущую команду пользователя. Окончание вычислений задается нажатием клавиши ETX.

Диалоговое формирование числа в R_0 требует большого числа нажатий на клавиши. Так, диалоговое формирование единицы в приводящемся примере задается последовательностью символов N, 1, ETX, где N — символ, задающий инициализацию диалогового формирования числа, 1 — собственно набор числа, ETX — задает конец диалогового формирования.

Пример сложения веса сотрудника с его ростом задается последовательностью символов: ↓, 4, →, 1, ↓, 5, +, 1. После этого на экране будет представлено изображение, показанное на рис. 5.

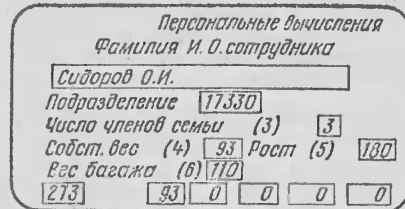


Рис. 5. Состояние экрана в режиме персональных вычислений

Рассмотрим пример персональных вычислений в другом режиме, который отличается от предыдущего только тем, что последовательность набираемых пользователем команд запоминается системой.

Пусть в этом режиме пользователем набрана следующая последовательность символов: ↓, 3, +, 1, →, 1, ↓, 4, +, 2, →, 2, ↓, 6, +, 3, →, 3, N, 1, ETX, +, 4, →, 4, ETX. При этом число членов семьи (↓, 3 — выбор числа из третьего показателя в R_0) складывается с содержимым R_1 (+, 1) и результаты засылаются в R_1 (→, 1). Аналогично собственный вес сотрудника и вес багажа складываются с содержимым R_2 и R_3 (результат в R_2 и R_3). К содержимому R_4 прибавляется 1 (N, 1, ETX, +, 4, →, 4).

Эту запомненную последовательность можно проинтерпретировать нужное число раз. Для этого достаточно соответствующее число раз нажать единственную клавишу И(1) — (исполни). Дважды запустив сформированную программу персональных вычислений для различных записей, получим в числовых регистрах (при условии, что в них были ну-

левые значения) суммы соответствующих показателей по двум записям.

В СПВ реализован режим, в котором объединены возможности поиска по образцу и исполнения запомненной последовательности вычислений, т. е. система осуществляет поиск и для каждой найденной (сравнившейся с образцом) записи выполняет запомненную последовательность вычислений. Пусть в поисковом образце задано подразделение (последовательность вычислений приведена при рассмотрении режима запоминания), тогда, осуществив обработку всего массива в режиме ПОИСК + ВЫЧИСЛЕНИЯ (предварительно необходимо загрузить все числовые регистры), пользователь получит в числовых регистрах соответствующие итоговые числа по подразделению. В регистре R_4 будет представлено общее число найденных записей, т. е. общее число сотрудников подразделения.

Использование перечисленных средств открывает широкие возможности для проектирования прикладных тем на основе подхода «макетирование плюс программирование без программиста». СПВ может быть реализована в виде демонстрационной модели. Для этого постановщик осуществляет следующую последовательность действий:

1. С помощью средств, аналогичных рассмотренным в начале статьи, проектирует бланк экранного документа для представления одной записи массива.

2. Формирует последовательность персональных вычислений над одной записью массива.

3. Формирует последовательность действий, необходимую для получения итогов по выбранным записям массива, которая состоит из формирования поискового образа, вызова на экран первой записи массива, занесения нулевых значений во все числовые регистры СПВ, запуска обработки массива в режиме ПОИСК + ВЫЧИСЛЕНИЯ.

Полученная в результате такой настройки демонстрационная модель предоставит пользователю (заказчику) возможность ведения информационного массива и осуществления запросов на выдачу итогов по всем отобранном записям. Процедура запроса может быть следующей: пользователь нажимает условную клавишу инициализации запроса, система предоставляет ему возможность сформировать поисковый образец и осуществляется автоматический подсчет итогов и их вывод на экран.

Возможна такая ситуация, когда демонстрационная модель удовлетворяет информационные потребности пользователя [3]. В этом случае она может непосредственно внедряться в практику.

Заключение. В восьмидесятых годах широкое распространение полу-

чили системы типа Visi Calc. Большинство таких систем фактически в том или ином виде реализует подход, рассмотренный в настоящей статье. При этом значительно повышается производительность труда при создании прикладных систем. Таблица прогресса программирования [7] позволяет сделать вывод о том, что производительность проектирования может повыситься более чем на порядок по сравнению с использованием языков высокого уровня.

Средства и трудозатраты на разработку

Год	Средства	Время разработки и внедрения, мес.
1980	Машинный язык	8
1965	Ассемблер	6
1970	Кобол	4
1975	Бейсик	1
1980	Visi Calc	2

По-видимому, именно такого повышения следует ожидать и в других случаях. Например, опыт использования генератора [4] подтверждает это.

По существу предлагаемый подход является определенным шагом в направлении автоформализации про-

фессиональных знаний [2] работников и в первую очередь работников сферы управления. Вместе с тем этот подход позволяет автоформализовать только «статическую» управления (в частности, форматы документов), оставляя без внимания его «динамику» (процедуры обработки).

Некоторые возможные пути автоформализации «динамики» управления известны — это, в частности, запоминание системой последовательности выполняемых пользователем «вручную» действий с экранными документами и последующее автоматическое повторение системой этих действий в аналогичных ситуациях. Но вопрос об автоформализации управленческих процедур требует отдельного большого разговора.

Телефон для справок: 3-32-82, г. Димитровград

ЛИТЕРАТУРА

1. Эпштейн В. Л. Автоматизация проектирования (вопросы методологии) / В кн.: Автоматизация проектирования систем управления. Вып. 4.— М.: Финансы и статистика, 1982.— С. 4—12.
2. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации.— М.: Наука, 1984.— 240 с.

3. Казменко С. В. Автоматизация разработки экранных документов: Препринт. НИИАР — 22 (668).— М.: ЦНИИАтоминформ, 1985.— 21 с.
4. Казменко С. В. Автоматизированная система разработки входных экранных документов (генератор программ форматного ввода информации АСУ) // Управляющие системы и машины — 1985.— № 2.— С. 88—93.
5. Казменко С. В., Шимбарова Л. А. Об использовании микроЭВМ в качестве интеллектуального устройства подготовки данных: Препринт. НИИАР—15(627).— Димитровград, 1984.— 12 с.
6. Дейт К. Введение в системы баз данных.— М.: Наука, 1980.— 460 с.
7. Emanielli G. Automazione degli uffici // Notiz. tech. AMMA.— 1984.— Vol. 39, № 7. 8.— P. 2—5. (P. 67. Организация управления.— М.: ВИНТИ, 1985, № 7, реф. 7.67.108).
8. Martin J. with research by Murch R. Application Development Without Programmers.— In: Savant Inst. Seminar Documentation by J. Martin. Carnioth: Savant Res. Studies, 1981.

Статья поступила 20 марта 1986 г.

УДК 681.3.068

А. Г. Григорьев

ДИАЛОГОВЫЙ ДИЗАССЕМБЛЕР ДЛЯ ЗАГРУЗОЧНЫХ МОДУЛЕЙ ОПЕРАЦИОННОЙ СИСТЕМЫ РАФОС

Во многих случаях при переработке или адаптации системных и прикладных программ имеются только их загрузочные или объектные модули, а исходные тексты отсутствуют. Разобраться в программе по восьмеричному дампу весьма трудно и зачастую невозможно, поэтому необходимо получить исходный текст. Для получения исходных текстов из машинных кодов применяются различного рода дизассемблеры. Дизассемблер бывает необходим также для анализа кодов программ, написанных на языке высокого уровня, при отыскании особо трудных ошибок (связанных, быть может, с ошибкой компилятора или подпрограмм исполняющей системы).

Полностью автоматические дизассемблеры имеют принципиальный недостаток — они не в состоянии отличить команды от данных, а данные различить по виду: символы, числовые и т. п. Сгенерированные ими тексты обычно очень трудны для понимания и практически непригодны для модификации и последующей трансляции. Кроме того, если в загрузочном модуле отсутствует информация по перемещению адресных констант (что, например, имеет место в системе RT-11 и многих других), то автоматически различить адресные и обычные константы не представляется возможным. Такая задача под силу лишь человеку.

Для получения исходных текстов на макроассемблере,

пригодных для модификации и последующей обработки транслятором MACRO-11, был разработан диалоговый дизассемблер DISASM. В него заложен принцип «ручного» определения типов полей и автоматической генерации текста. Дизассемблер обладает следующими возможностями:

обрабатывает загрузочные модули типов SAV, REL и LDA. Модули типа SAV могут быть оверлейными и включать различные версии оверлейного драйвера (кроме виртуального); LDA — загружены с магнитной ленты, имеющей формат TMOС. Из каждого сегмента оверлейного модуля генерируется отдельный файл с текстом. Может быть сгенерирован полный текст модуля или сегмента либо текст отдельных фрагментов;

по команде оператора запоминает промежуточные результаты работы в файле и может впоследствии их восстановить;

генерирует мнемоники всех команд, обрабатываемые транслятором MACRO V05.00, и директивы определения данных .ASCII, .ASCIZ, .RAD50, .WORD, .BYTE, .BLKW, .BLKB;

генерирует непосредственные операнды команд в виде восьмеричного или десятичного числа, ссылки на метку, слова в коде RADIX-50 (префикс $\wedge R$), имени кода операции (например, $\# MOV$), одного или двух символов, причем печатные символы изображаются с одинарным или двойным апострофом, а управляющие символы — соответствующими именами;

генерирует директиву .WORD, операнды которой могут иметь вид десятичного или восьмеричного числа, ссылки на метку, одного или двух символов, смещения до метки в виде $A-(+2)$;

генерирует директиву .BYTE, операнды которой могут иметь вид десятичного числа, восьмеричного числа, символа;

генерирует обращения к макрокомандам (.CHAIN, .CLOSE, .DATE, .EXIT, .HERR, .HRESET, .LOCK, .PRINT, .PURGE, .RCTRLO, .SERR, .SETTOP, .SRESET,

TTINR, TTOUTR, .TTYIN, .TTYOUT, .UNLOCK, .WAIT, .MWAIT, .ABTIO, .RSUM, .SPND, .TLOCK), аргумент которых находится в R0 или отсутствует. Для макрокоманд, аргументы которых засылаются в стек (.CSIGEN, .CSISPC, .DSTAT, .FETCH, .QSET, .RELEASE), аргументы сгенерировать трудно, поэтому в тексте помещаются лишь комментарий с их названием; при ссылках на ячейки системной области связи (JSW, USERTOP, USERPC, USERSP и пр.), а также на регистры внешних устройств (TT, LP, PC, DX, MT, MM, RK, RM, DP, PSW, диспетчера памяти) генерирует их символические имена. Значения имен регистров устанавливаются прямым присваиванием в начале текста; разделяет программу на блоки локальных меток и различает пять типов меток: локальные, метки при данных и метки при командах, входы в подпрограммы, глобальные метки.

Обработка модуля начинается с «ручного» просмотра его дампа на экране дисплея, при этом каждому слову или байту загрузочного модуля присваивается тип, например команда, поле в коде ASCII, восьмеричное или десятичное число и т. п. Дамп на экране представляется в байтовой или пословной форме. При выводе в байтовой форме выводится восьмеричные коды байтов и их символичные эквиваленты, причем управляющие символы заменяются их мнемоническими обозначениями. Рядом с восьмеричным изображением словом или байтом указывается тип. При представлении дампа по словам на экран выводится восьмеричное изображение, под ним его символичный эквивалент и эквивалент в коде RADIX-50. Символьное изображение выводится также в правой части экрана в виде полосы (как на обычных дампах). Нажатием некоторых клавиш оператор указывает тип одиночных слов и байтов или их полей. Начало обрабатываемого поля устанавливается отдельной клавишей и изображается на экране точкой. Тип, присвоенный некоторому слову или байту, можно изменить на другой или вообще отменить присвоение (текст будет сгенерирован лишь из тех кодов, которым присвоен тип).

При просмотре дизассемблер управляется клавишами основной и дополнительных клавиатур. Клавиши ", ' , >, A, B, D, I, K, M, O, R, V, W, Z указывают тип поля либо отдельного слова или байта следующим образом:

- " — слово, содержащее два символа;
- ' — байт, содержащий один символ;
- > — слово, содержащее смещение (подобное индексному слову при относительной адресации);
- A — поле символов в коде ASCII;
- B — поле байтов некоторого типа. При его обработке тип ненулевых байтов запрашивается отдельно;
- D — поле слов или байтов, которые в тексте будут представлены в виде десятичных чисел (если курсор находится на байтовом изображении, то поле считается байтовым, а если на пословном, то считается, что это слова);
- I — последовательность команд;
- M — слово, содержащее адрес;
- O — поле слов или байтов, которые в тексте будут представлены в виде восьмеричного числа;
- R — поле слов, содержащих указательные символы в коде RADIX-50. Из него генерируются директивы .RAD50;
- V — два слова, образующие вектор прерывания. Первое слово получает тип «адрес», второе слово — тип «восьмеричное число»;
- W — поле слов некоторого типа. При его обработке тип ненулевых слов запрашивается отдельно;
- Z — поле нулевых слов или байтов, из которого будет сгенерирована директива .BLKW или .BLKB.

Клавиша пробела отменяет присвоение типа полю или отдельному слову или байту (тип их становится неопределенным). Клавиша @ перемещает курсор на слово или байт, адрес которого находится в текущем слове, Клавиша S выполняет поиск ранее введенного об-

разиа. Клавиши CV/W и CV/R восстанавливают экран после сбоя. При нажатии клавиши ? последовательно выводятся две справочные таблицы.

Клавиши дополнительной цифровой клавиатуры выполняют перемещение курсора на слово, которое было обработано последним, установку начала обрабатываемой области (оно изображается на экране точкой), установку режимов пословного и байтового изображения на экране, переход к предыдущему или к следующему оверлейному сегменту, поиск слова или байта, которому еще не назначен тип, ввод образца с последующим поиском, переход к диалогу для записи результатов работы в файл, переход к диалогу для генерации текста.

Клавиши функциональной клавиатуры выполняют перемещение курсора вперед/назад на слово или байт, 16 (20₈) байт и на 160 (240₈) байт, перемещение курсора на начало или конец оверлейного сегмента или загрузочного модуля, перемещение курсора на «точку».

При нажатии клавиш указания типа обрабатывается область от «точки» до текущего положения курсора. После обработки некоторого поля «точка» автоматически устанавливается на слово или байт, следующий за ним. Если курсор установлен на «точке», то большинство команд дизассемблера обрабатывает одно слово или байт, но при нажатии некоторых клавиш обрабатывается сразу целое поле. Например, нажатие клавиши A (которой указывается поле в коде ASCII) вызывает обработку последовательности символов до нулевого байта или байта с кодом 200. При нажатии клавиши Z (.BLKW, .BLKB) обрабатывается поле до первого ненулевого байта.

При нажатии клавиши I (указание последовательности команд) дизассемблер рассматривает последовательность слов и присваивает им признак команды до тех пор, пока не встретит одну из команд RTS, JSR, TRAP, BPT, IOT, RTI, RTT, HALT, BR, JMP, EMT 350 (макрокоманда .EXIT), EMT 376 (системная ошибка) с регистром, отличным от счетчика команд PC, а также команду, явно изменяющую счетчик команд (например, ADD RO, PC). Если будет встречена одна из команд ADD, SUB, CMP, MOV с непосредственным операндом, то запрашивается вид, в котором он должен быть представлен, после чего обработка последовательности продолжается. Непосредственным операндам команд BIS, BIT, BIC и одноадресных команд автоматически назначается представление в виде восьмеричного числа. Следует заметить, что этот тип может быть изменен оператором на любой другой.

Например, в системных программах часто встречается последовательность команд следующего вида:

```
TST      (PC)+
10x:    SEC
        RETURN
```

Если в этом случае непосредственному операнду команды TST явно не присвоить тип «команда», то будет сгенерирован такой текст:

```
TST      (PC)+
10x:    .WORD 261
        RETURN
```

Индексные слова команд могут быть либо адресами, либо числами. Если индексирование производится относительно счетчика команд, индексное слово всегда считается смещением, а если относительно указателя сте-

ка — то числом. В прочих случаях, если индексное слово меньше 1000, или больше верхнего адреса программы, оно считается числом, в противном случае оно обрабатывается как адрес. Если в программе используются индексы, превышающие 1000, то индексным словам нужно присваивать тип «восьмеричное число» явно. Такой случай встретился при дизассемблировании системной программы DUP.

После того, как тип присвоен всем (или требуемым) кодам загрузочного модуля, нажимается соответствующая клавиша на дополнительной цифровой клавиатуре, запускающая автоматическую обработку, состоящую из нескольких проходов. Далее в диалоге запрашиваются имена выходных файлов и генерируется текст. При автоматической обработке выполняются следующие действия:

- определяются границы блоков локальных меток;
- определяются точки, на которые имеются ссылки, и типы меток;
- производится нумерация меток (для каждого типа — своя последовательность номеров);
- выделяются команды, которые можно заметить макрокомандами.

Границы блоков локальных меток определяются на первом проходе по ссылкам из команд ветвления и SOB, смещение в которых не превышает 300₈ по модулю (т. е. от -192 до +192). Предполагается, что команды ветвления со смещением от -400 до -300₈ и от +300 до +376 используются преимущественно для переходов из одного локального блока в другой. Для определения границ выделяются области перекрытия ссылок. Когда текущая точка просмотра отстоит от начала первой такой области более чем на 300₈, выполняется поиск допустимой границы LSB, которой может быть одна из команд BR, JMP, RTS, RTI, RIT, HALT, EMT 350, или данные. Если такая точка найдена, ей присваивается признак границы точки. После этого первая область перекрытия ссылок исключается, а ее верхний адрес запоминается в качестве границы для поиска следующей границы LSB. Как было установлено при эксплуатации дизассемблера, такой алгоритм в большинстве случаев дает вполне удовлетворительные результаты.

Регистрация меток производится на втором автоматическом проходе. При этом определяются точки, на которые имеются ссылки, и тип меток (локальные метки, метки при командах, метки при данных, входы в подпрограммы, глобальные метки). Определяется также, к каким регистрам внешних устройств имеются ссылки.

Локальными метками считаются те, на которые есть ссылки только в данном блоке локальных меток, причем на них ссылаются только команды ветвления со смещением, не превышающим 300. Эти метки нумеруются через 10 в пределах каждого локального блока.

Метки, на которые есть ссылки из других оверлейных сегментов, считаются глобальными. Дизассемблер генерирует правильные метки как при обращении «вверх» или «вбок» (к сегменту того же уровня) через таблицу оверлейных вызовов, так и при обращении «вниз» в корневой сегмент. При обращении «вверх» (такой случай встретился в MACRO-11 и в K52) или «вбок», как это было в программе PIP, не через таблицу вызовов, текст может быть сгенерирован неправильно. Обработка обращения «вниз» (из одного сегмента к оверлейному более низкого уровня) производится с учетом того, какой именно сегмент из той области, в которую была ссылка, вызывал сегмент, ссылка из которого обрабатывается. При обработке обращения «вверх» не через оверлейный драйвер для каждой оверлейной области упоминается номер сегмента, к которому было последнее обращение через оверлейный драйвер.

Метки, на которые ссылаются команды JSR (но не из других сегментов), считаются входами в подпрограммы.

В тексте глобальные метки получают имена Gnnpp, метки при данных — Dnnpp, метки при командах —

Inppp и метки при подпрограммах — Snppp, где pppp — десятичное число (номер метки). Если имеется ссылка на старший байт слова, то метка устанавливается на младший байт, а ссылка в тексте генерируется в виде A+1. Ссылки на нечетный байт поля байтов обрабатываются обычным образом.

Третий проход. Выполняется нумерация меток. Метки каждого типа нумеруются независимо от других с шагом 10, причем глобальные метки имеют сквозную нумерацию в пределах всего оверлейного модуля, локальные — в пределах каждого блока, а прочие — в пределах оверлейного сегмента. При такой нумерации число меток каждого типа может достигнуть 32767 (после метки с номером 65530 будет идти метка с номером 4, и так далее, пока не будут исчерпаны все четные числа). Этого более чем достаточно.

Четвертый проход. Перед генерацией выделяются последовательности команд, которые можно заметить макрокомандами. Для этого отыскиваются команды EMT, проверяются команды, которые засылают аргументы, и, если макрокоманда может быть сгенерирована, первая ее команда помечается специальным знаком.

После выполнения всех проходов у оператора запрашивается имя файла (или несколько файлов для сегментов оверлейного модуля) и выполняется генерация текста. Текст генерируется лишь из тех слов или байтов, которым назначен тип. В начало генерируемого файла помещаются комментарии, в которых указывается имя загрузочного модуля, номер сегмента и номер оверлейной области для оверлейных сегментов. Далее в текст включаются директивы .ENABL, .MCL, .MCALL, .PRINT, макрокоманды .CTRLT и .SCMDF. Первая директива разрешает автоматический вызов макрокоманд, вторая директива нужна потому, что макрокоманда .PRINT не может быть вызвана автоматически (имеется такая директива). Макрокоманда .CTRLT определяет мнемонические имена управляющих символов, а макрокоманда .SCMDF — мнемонические имена ячеек системной области связи. Далее в текст включаются директивы прямого присваивания значений имен регистров внешних устройств, на которые имеются ссылки. Если текст генерируется из кодов, расположенных ниже начального адреса указателя стека, то в него включается директива .ASECT, а перед генерацией текста из кодов, расположенных выше начального адреса стека, добавляется директива .PSECT. Через каждый 400₈ байт в генерируемый текст помещаются комментарии вида

; . = ppppp

где ppppp — текущий адрес. Если некоторой области загрузочного модуля не присвоен тип, то она пропускается и в выходной файл помещается директива вида

. = . + pppp

где pppp — размер пропускаемой области.

Известно, что команды BCC и BHIS имеют код 103000, а команды BCS и BLO — код 103400. Одна из двух мнемоник выбирается исходя из того соображения, что команды BCC и BCS обычно используются после команд ADD, JSR, EMT, прерываний, сдвигов, а команды BLO/BHIS — после команд CMP, CMPB, SUB. Учитываются команды установки режимов процессора плавающей запятой, благодаря чему генерируются мнемоники команд одинарной и двойной точности.

Генерируемый текст соответствует исходным кодам даже при нетрадиционном или некорректном применении команд. Например, если имеется ссылка на второе или третье слово команды, являющееся непосредственным операндом или абсолютным адресом (некоторые программисты довольно часто используют такой прием), то генерируется конструкция

MOVEN (PC)+R1, D20; WORD 10

Если метка должна быть у индексного слова, то дизассемблер вынужден генерировать директиву .WORD, например:

```
D50: .WORD MOV+6701
      .WORD D30-(.+2)
```

При генерации директив .ASCII и .ASCIZ дизассемблер автоматически выбирает символ-ограничитель строк из следующего ряда: ', ", /, \, !, *, ^, %, #, @, ?, [], —, +. Если в символьной строке встречается апостроф то в качестве ограничителя используются двойные кавычки, а если в ней есть и апостроф, и кавычки, то используется слэш и т. д. Область, из которой генерируется одна директива, ограничивается либо байтом с кодом 0 или 200, либо некоторой длиной строки программы (около 60 символов), либо прекращается, когда кончатся символы данного типа, встречается управляющий символ или байт, при котором должна быть метка, или исчерпаны все предусмотренные символы-ограничители. Если строка оканчивается нулевым байтом, то генерируется директива .ASCIZ, а если оканчивается байтом 200, то — директива .ASCII, которая оканчивается восьмеричным числом 200 в угловых скобках. Когда встречается управляющий символ, то генерируется директива .BYTE, например:

```
D10: .ASCII 'COMPILATION STATISTICS'
      .BYTE CR, LF, CR, LF
      .ASCII 'Compilation time: '
D20: .ASCIZ 'HH.MM.SS'
```

Когда в этом примере встретился байт с кодом 15, началась генерация директивы .BYTE, а когда встретился байт, при котором должна быть метка, началась генерация новой директивы .ASCIZ. Подобным образом генерируется директива .RAD50. Если в этом случае встречается слово, не представимое в коде RADIX-50, то генерируется директива .WORD.

При генерации директив .WORD и .BYTE число operandов ограничивается длиной строки и наличием меток (когда встречается слово или байт с меткой, то начинается генерация следующей директивы). При генерации директив .BLKW и .BLKB сначала подсчитывается, какого размера область может быть описана. Если эта область включает менее одного слова или байта, то вместо, например, .BLKW1 генерируется .WORD.

Ниже приведен фрагмент текста, сгенерированного из корневого сегмента редактора K52 (комментарии добавлены автором):

```
.ENABL LSB
; Вставка символа с обработкой режима WRAP
S100: CALL G2230 ; Вставить символ в файл
      BCS 40x ; нет места
      TST B410 ; Установлен режим WRAP ?
      BEQ 40x
      MOV R0, -(SP)
      CMP #', R0 ; Вставляется печатный символ ?
      BHI 30x ; управляющий
      INC D270 ; увеличить номер позиции на 1
      BIT #4, G360 ; номер позиции правильный ?
      BNE 10x
      CALL B270 ; вычислить номер заново
      MOV R2, D270 ; новое значение
100: CMP G420, D270 ; не превысил заданного ?
      BHIS 20x
      CALL B140 ; выполнить перенос
      CLC
      BR 100 ; цикл в себе
```

```
20x: BIS #4, G350 ; признак наличия номера позиции
30x: MOV (SP)+, R0
40x: RETURN
      .ENABL LSB
      B140: CLR B430
      MOV B6730, R2
```

При работе дизассемблера обрабатываемый загрузочный модуль, признаки полей и номера меток хранятся в рабочем файле. Каждому слову модуля соответствует одно слово признаков и одно слово с номером метки. Если слово содержит информацию байтового вида (например, в коде ASCII), хранятся признаки каждого байта. Если оба байта слова должны иметь метку, номер метки хранится лишь для младшего байта. Чтобы определить номер метки для старшего байта, приходится просматривать номера предшествующих меток. Так как скорость генерации определяется преимущественно скоростью доступа к рабочему файлу, то для ее повышения программно имитируется кэш-память.

Содержимое рабочего файла можно записать в постоянный файл, а затем восстановить. Это используется для обработки программы по частям, а также для исправления ошибочно указанных типов полей. Например, если в сгенерированном тексте слово представлено как адрес, хотя употребляется как пара символов, то можно вновь восстановить рабочий файл, внести необходимые изменения и сгенерировать новый текст.

Дизассемблер написан на языке макроассемблера MACRO-11. Размер исходного текста около 7000 строк, загрузочный модуль имеет размер 68 блоков (34К байт) но в памяти занимает лишь 12К байт, а остальная доступная память используется в качестве буфера рабочего файла. Дизассемблер работает в операционной системе АДОС * на дисплее 15ИЭ-00-013 в режиме эмуляции VT52. Скорость автоматической обработки (автоматические проходы и генерация) составляет примерно 1—2 блока (512 байт) загрузочного модуля в секунду и существенно зависит от скорости работы устройства, на котором находится рабочий файл. Ручной просмотр требует в среднем 1 мин/блок.

Несмотря на то, что данный дизассемблер рассчитан на обработку только загрузочных модулей, с его помощью обрабатывались и объектные модули (но при этом возникали затруднения с внешними ссылками). Для этого их преобразовали в загрузочный модуль. Таким путем были переработаны транслятор Фортран и подпрограммы его исполняющей системы.

Дизассемблер применялся также в качестве «экранного дампа» для просмотра загрузочных модулей и частично дизассемблирования, что позволило найти места для корректировок «заплатами».

* Григорьев А. Г. Адаптированная операционная система АДОС для СМ ЭЕМ // Микропроцессорные средства и системы. См. наст. номер, с. 23.

Телефон для справок: 996-252,
Томск

Статья поступила 16 сентября 1986 г.

А. Г. Григорьев

АДАптированная ОПЕРАЦИОННАЯ СИСТЕМА АДОС ДЛЯ СМ ЭВМ

Для работы с системой команд микроЭВМ ДВК «Электроника 60» и т. п. широко используются операционные системы РАФОС, ФОДОС, ФОБОС, совместимые с RT-11. Общий недостаток их — выдача сообщений на английском языке. Кроме того, в этих операционных системах текстовую информацию приходится представлять в сокращенном наборе КОИ-7, включающем только прописные буквы. Это вызвано тем, что все системные программы ориентированы на работу с 7-битовыми кодами символов. Между тем получили распространение устройства, позволяющие работать с полным набором символов, такие, как терминалы 15ИЭ-00-013 и печатающие устройства D-180. Отсутствие программной поддержки препятствует полному использованию их возможностей.

Разработана операционная система АДОС (совместимая с RT-11 версии V05.00), в которой для хранения текстов используется код КОИ-8. При этом применяется диалоговый дизассемблер [1]. От RT-11 (V05.00) АДОС отличается следующим:

обмен с терминалом стал производиться в коде КОИ-7 (полный набор с использованием переключающих символов S0 и S1). Так как для внутреннего представления текстов используется код КОИ-8, то при вводе и выводе производится соответствующее преобразование;

так как символ СУ/0 (код 17) используется как символ (ЛАТ) то вместо него отмена вывода на терминал в АДОС выполняется символом СУ/Е (код 5);

для работы с терминалами, не отображающими символы нижнего регистра, введена команда SET TT LC/NOLC. Команда SET TT LC переводит систему в режим обмена с терминалом в полном наборе символов, а команда SET TT NOLC устанавливает работу в сокращенном наборе символов. При работе в сокращенном наборе символы, вводимые с клавиатуры, преобразуются в русские и латинские символы верхнего регистра по КОИ-8, а при выводе русские буквы нижнего регистра переводятся в русские буквы верхнего регистра (твердый знак «ъ» преобразуется в апостроф). Латинские буквы нижнего регистра отображаются в виде русских букв верхнего регистра, что позволяет выполнять в этом режиме программы, разработанные для операционной системы РАФОС;

введена команда SET TT IE1513, при выполнении которой на дисплей 15ИЭ-00-013 посылается символ с кодом 27 (СУ/W), переводящий его в режим частичной эмуляции команд терминала VT52, и устанавливаются режимы TAB, SCOPE, LC;

для повышения скорости обмена с терминалом в процедуру обработки прерываний от вывода включена предварительная подготовка символа, заключающаяся в том, что очередной символ извлекается из буфера и преобразуется в КОИ-7 во время вывода предыдущего символа, а не после того, как он будет выведен;

для снижения затрат времени на блокирование и разблокирование задачи по условию наличия места в буфере вывода, разблокирование выполняется только после вывода половинного буфера, а не одного символа;

в режиме SCOPE при вводе управляющего символа СУ/У выполняется стирание строки с экрана дисплея.

Переработка большинства системных программ заключалась в том, что сообщения были просто переведены на русский язык, а при работе с текстами отменена очистка старшего бита символов.

В программу DIR, кроме того, добавлена выдача сообщения о размере наибольшей свободной области, имеющейся на диске, а также формирование грамматически правильных окончаний слов, следующих после чисел. Например:

```
16-Май-86
HELLO .TXT      3 16-Май-86 (ПУСТО)    60
DIS .DOC       25 13-Май-86 (ПУСТО)    26
ADOS .DOC      27 16-Май-86 ADOS .DPK   30
(ПУСТО) 315
```

4 файла, 85 блоков

Имеется 401 свободный блок

В наибольшей области 315 блоков

В систему введен новый формат справочных библиотек, используемый программой HELP и редактором K52. Для этого программа LIBR была дополнена соответствующей процедурой. Имена разделов такой библиотеки, содержащие до восьми символов, хранятся в оглавлении в коде КОИ-8. Вместе с именем раздела хранятся адреса его начала и конца. Справочная библиотека получает тип «HLB», а для ее формирования в командной строке указывается ключ /H.

Программа вывода справок HELP модифицирована для работы с описанным форматом справочных библиотек. Кроме того, для нее можно задавать имя библиотеки.

В программу LINK добавлена проверка контрольных сумм записей объектных модулей.

В программе MACRO исправлены некоторые ошибки. Например, из-за неправильного употребления макрокоманды .PURGE при ошибке в файле перекрестных ссылок вместо канала 12 очищался канал 0, в который выводился объектный модуль. Из-за другой ошибки не всегда правильно выводились данные об использовании рабочего файла. В исходной версии при хранении макроопределений в рабочем файле старший бит являлся маркером служебных байтов. Так как при использовании кода КОИ-8 этот бит занят, пришлось перед всеми служебными байтами вставлять в качестве флажка нулевой байт. Внесено изменение, запрещающее директивам .NCHR, .NTYPE, .NARG присваивать значение счетчику адреса. Добавлен новый оператор .SWITCH с синтаксисом:

.SWITCH имя

Имени присваивается значение, указанное с ключом /SWITCH:n в командной строке. Может быть указано несколько значений, в этом случае для доступа к ним записывается несколько операторов SWITCH. Этот оператор полезен, если необходимо часто изменять параметры трансляции для получения различных вариантов программ.

В настоящее время испытывается версия транслятора, в которую для повышения быстродействия внесены следующие изменения:

размещение подпрограмм по оверлейным сегментам, что значительно сократило инициализацию и завершение трансляции, а также ускорило автоматический вызов макроопределений;

текущий словарь локальных меток сделан резидентным (резервируется память под 128 меток, чего вполне достаточно) с использованием двоичного поиска;

применены резидентные подсловари имен меток и команд (макрокоманд) с двоичным поиском. Для каждого словаря резервируется память под 64 имени. Поиск выполняется сначала в резидентном словаре, а только затем в рабочем файле. Если имя найдено в файловом словаре, то оно помещается в резидентный словарь. Для вытеснения имен из резидентного в файловый словарь использован алгоритм LRU;

для файлового словаря использована структура сбалансированного двоичного дерева с применением алго-

ритма Адельсона-Вельского и Ландиса [2], в то время как в исходной версии применялось хэширование, из-за которого рабочая память использовалась неэффективно (оставалось много пустых мест).

В системную макробibliothek были внесены изменения, подобные тем, которые сделаны в РАФОС, например добавлена возможность употребления аргумента AREA-BLOCK, а также макроопределения «DIR», CTRLT, JSWDF, SCMDP и т. п. Модифицировано макроопределение PRINT — добавлены аргументы MSG- (сообщение) и CRLF-NO, позволяющие задавать текст непосредственно в макрокоманде. При указании аргумента MSG-(сообщение) генерируются следующие команды (уникальность номеров локальных меток обеспечивается транслятором):

```

JSR      /0,30000x
„ASCIZ  \сообщение\
„EVEN
30000x: EMT      ^D351
MOV      (6.)+,/0
  
```

Если задано CRLF-NO, то вместо директивы .ASCIZ генерируется директива .ASCII с конечным байтом (^ 0200).

В программе CREF исправлена ошибка, из-за которой номера строк больше 1000 печатались неправильно.

Транслятор Фортран модифицирован для обработки исходных текстов в коде КОИ-8. В нем также переведены сообщения об ошибках. Подпрограммы исполняющей системы тоже переработаны соответствующим образом, причем, несмотря на то, что русские тексты сообщений имеют больший размер, нежели английские, размер модуля ERRS не увеличился благодаря сборке сообщений из общих фрагментов.

Существенной переработке подвергся экранный редактор текстов К52. Заново написаны процедуры отображения экрана, управления текстами (вперед на символ, назад на символ, вставить символ, стереть символ), перенумерации локальных меток по команде LOCAL, добавлены новые возможности, а также изменено расположение некоторых процедур по оверлейным сегментам.

В исходной версии процедура отображения экрана работала довольно медленно, особенно если на экране много текста. Это заметно ощущалось при работе на процессоре с низким быстродействием. В ней сначала формировался образ второй над курсором строки, затем этот образ сравнивался со строками старого образа экрана, находилась такая же строка, и исходя из этого определялось, на сколько строк надо сдвигать экран. Затем формировался новый образ экрана и выводились изменившиеся места. Такой алгоритм определения сдвига экрана часто срабатывал не лучшим образом. Кроме того, курсор постоянно находился в 16-й строке экрана, что не всегда удобно.

В новом варианте для определения оптимального сдвига экрана процедуры управления текстом постоянно отслеживают изменение числа разделителей строк (пар символов BK+PC и одиночных символов BK, PC, PF, VT) над и под курсором, а процедура отображения хранит номера непробельных строк. Курсору теперь разрешено находиться в 6...18 строках экрана (при условии, что сверху и снизу достаточно текста). Новая строка, где будет находиться курсор, определяется как сумма номера строки, где курсор находился раньше, с изменением числа разделителей строк сверху от курсора. Вычисленный номер ограничивается 6...18 строками. Далее проверяется, достаточно ли текста снизу и сверху, чтобы курсор мог находиться на вычисленной строке, если недостаточно, то номер строки

корректируется. По вычисленному номеру строки и по изменению числа разделителей строк над и под курсором определяется, на сколько строк следует сдвинуть части экрана, находящиеся над и под курсором. Так как дисплей не выполняет вставку и исключение строк, то приходится сдвигать экран целиком. Для этого анализируется, при сдвиге какой части экрана больше строк не потребуют изменений.

Описанный алгоритм отображения оказался более эффективным, чем исходный. Например, при вставке строки в начало файла редактор исходной версии выводит заново нижнюю часть экрана, а новый редактор сдвигает его на одну строку вниз. Быстродействие процедуры отображения повысилось еще и благодаря тщательной оптимизации всех циклов, в результате чего на мини ЭВМ «Электроника 100/25» формирование образа экрана (без учета вывода на терминал) требует по 50 мкс/байт отображаемого текста плюс 50 мс независимо от содержимого экрана. Поэтому даже при работе на маломощных машинах не ощущается прежних неудобств. В новой процедуре символы вертикальной таблицы, перевода формата и одиночные символы перевода строки и возврата каретки отображаются символами V, F, L, R соответственно в 79-й позиции строки. Изменена форма изображения длинных строк.

Переработаны процедуры управления текстом, что повысило их быстродействие примерно в 1.5...2 раза. Они, кроме того, перенесены в корневой сегмент, что исключило затраты времени на работу оверлейного драйвера.

В исходном варианте процедура перенумерации локальных меток (команда LOCAL) была спроективана весьма неудачно, так как выполняла множество переходов по тексту. Например, обработка фрагмента, содержащего около 50 локальных меток и примерно 100 ссылок на них, требовала 10...20 мин. Написанная новая процедура выполняется за два прохода: на первом формируется словарь, на втором все метки заменяются новыми. Это увеличило быстродействие при обработке больших блоков локальных меток чуть ли не на два порядка. При определении границ блока локальных меток, как и в исходной версии, учитываются операторы .ENABL LSB и .DSABL LSB, простые метки и символы перевода формата. Кроме того, при действии оператора .DSABL LSB учитываются директивы .PSECT, .ASECT, .CSECT, .RESTO. Корректно обрабатываются следующие конструкции:

```

„ASCII /15x/      ; пропускается
„RAD50 /20x/      ; пропускается
„WORD  "5x        ; пропускается
MOV     #^R 6x,R0  ; пропускается
; 20x - пропускается
MOV     ^D10x,R1   ; обрабатывается
  
```

Метки распознаются не по виду (как в исходной процедуре), а по номеру, как в трансляторе, например метки вида 010 □ и 10 □ считаются одинаковыми. Обнаруживаются ссылки на неопределенные и повторяющиеся метки.

Кроме возможностей чтения и записи дополнительных файлов, имеющихся в исходной версии, добавлена возможность экранного просмотра двух дополнительных файлов и раздела справочной библиотеки. Для доступа к файлам выдается команда READ. Дополнительные файлы можно использовать для просмотра файлов листингов параллельно с редактированием исходных текстов, из них можно переносить различные фрагменты в редактируемый файл (сами они не изменяются). Обращение к разделам справочной библиотеки производится командой ?. По умолчанию ее имя формируется из типа

редактируемого файла, например если редактируется файл с типом .MAC, то будет использоваться библиотека SY:MAC.HLB. Указать другое имя можно командой HELP. Раздел библиотеки просматривается так же, как и дополнительный файл. Для перехода от одного файла к другому нажимаются клавиши <GOLD> p <GOLD> <F>, где p — номер файла (0 — основной файл, 1 и 2 — первый и второй дополнительные файлы, 3 — раздел справочной библиотеки). Такой переход может выполняться в любой последовательности.

Кроме перечисленных, внесены различные мелкие исправления, такие, как установка режима NOPAGE при запуске редактора и восстановление прежнего режима при выходе, восстановление слова TTCNFG в мониторе при аварийном завершении работы (редактор изменяет его при запуске). Чтобы можно было работать на дисплее 15ИЭ-00-013, из редактора убраны последовательности псевдографических символов и использованы режимы повышенной яркости.

Разработана новая версия программы формирования текстов DOC V06.02. В отличие от программы DOC V05.00, поставляемой с операционной системой РА-ФОС-2, она полностью написана на макроассемблере, что повысило ее быстродействие в 5 раз и уменьшило размер загрузочного модуля до 29 блоков. DOC V06.02 обрабатывает тексты в коде КОИ-8. Совместимость с версией V05 в основном сохранена, изменены лишь способ запуска, некоторые ключи командной строки, формат operandов директив MRR, MRL, изменена обработка директив .OVP и .JMP без аргумента, добавлены директивы .SUB — подчеркивание строк, .SKP — пропуск номеров страниц, .CNG — формирование листа регистрации изменений, .SPC — определение символа для резервирования пробелов в форматированном тексте. В новой версии при обработке файла директивой .CPL символы перевода формата вызывают переход на новую страницу, пропуск символа в директиве .OVP отменяет со-

ответствующее действие, директива .JMP действует лишь на форматированный (словесный) текст, а директива .JMP без аргумента устанавливает пропуск, заданный ключом /J:p в командной строке при запуске программы. Новая версия может формировать страницы размером до 100 строк. Это дает возможность применять плотность 3 строки на дюйм (при печати через строку это соответствует печати через два интервала на пишущей машинке). Имеется также ключ установки длины строки при запуске программы, что позволяет использовать печатающие устройства с плотностью 12 символов на дюйм. Программа DOC была использована, в частности, для форматирования текста рукописи данной статьи, а исходный текст ее был набран с помощью новой версии редактора K52.

Разработана новая программа установки текущей даты при начальной загрузке. В отличие от программы DATIME, имеющейся в RT-11, она запоминает дату предыдущей загрузки, поэтому если дата осталась неизменной, то достаточно нажать BK, а если увеличилась на один день — то +BK.

В настоящее время работа над системой продолжается, и в варианте, который будет поставляться, возможно, будут и другие отличия и добавления.

ЛИТЕРАТУРА

1. Григорьев А. Г. Диалоговый дизассемблер для загрузочных модулей операционной системы РА-ФОС // Микропроцессорные средства и системы.—См. наст. номер, с. 19.
2. Кнут Д. Искусство программирования для ЭВМ. Кн. 3. Сортировка и поиск.—М.: Мир, 1978.

Статья поступила 16 сентября 1988 г.

УДК 681.3.06

В. Г. Доломанов, А. В. Канофьев

СТРУКТУРНЫЕ ОСОБЕННОСТИ ИНТЕРПРЕТИРУЮЩЕЙ СИСТЕМЫ БЕЙСИК/РАФОС

Бейсик — одна из наиболее популярных систем программирования на мини- и микропроцессорной технике. Простота конструкций, достаточно мощные вычислительные средства совместно с удобствами, предоставляемыми Бейсиком как интерпретирующей системой, делают его практически незаменимым для работы в режиме «калькулятора», построения несложных имитационных моделей обработки статистических данных, запрос-ответных систем и т. д. Дистрибутивный комплект интерпретатора, поставляемый ЦФАП (г. Калинин), позволяет пользователям получить для себя специализированную версию с учетом требуемого набора функций, используемой ЭВМ, периферийных устройств: расширить функциональные возможности интерпретатора путем подключения дополнительных подпрограмм пользователя.

Система генерации интерпретатора

Создание рабочей версии системы Бейсик, паделенной требуемым набором функций, производится в два этапа: генерация интерпретатора, запуск и установка его для работы. Генерация интерпретатора производится с помощью программы генерации SUCNFG, которая на основании ответов пользователя создает командный файл для компоновки интерпретатора из объективных модулей, входящих в дистрибутивный носитель.

В момент запуска интерпретатора производится его дополнительная настройка. Она заключается в последовательном уплотнении библиотеки встроенных функций интерпретатора. Это обеспечивается с помощью настроечной таблицы, каждый элемент которой описывает соответствующую функцию:

Номер слова	Содержательный смысл
-------------	----------------------

- | | |
|---|--|
| 0 | Адрес строчек имени функции либо 0 для конца таблицы |
| 1 | Адрес начала программного кода функции |
| 2 | Адрес конца программного кода функции |
| 3 | Адрес ссылки на модуль TABLES |
| 4 | 0 (ограничитель элемента) |

В результате работы алгоритма уплотнения формируются следующие параметры: адрес новой верхней границы свободной памяти, откорректированная таблица ссылок на область размещения у кодов функций. Уплотнение библиотеки функций обеспечивается позиционно-независимым методом кодирования их алгоритмов. Сгенерированный и настроенный модуль интерпретатора размещается в оперативной памяти последовательными секциями, взаимосвязанными друг с другом управляющими и адресными ссылками. Каждая секция объединяет в себе часть функций, используемых интерпретатором. Последовательное распределение секций в сторону возрастания адресов и их примерное функциональное наполнение следующее:

PRECT	Функциональное наполнение
BASPCN	Наименование версии системы
BASRCO	Базовые функции обработки текста, модули синтаксического анализа, символьные преобразования
BASOTS	Численная математика. Тригонометрические функции
BASSCO	Средства ввода-вывода, работа с файлами, управление терминалом
BASXCO	Интерпретирующий процессор. Идентификация и исполнение основных операторов языка
BASXRO	Работа с данными. Чтение, запись переменных, анализ допустимости их значений
BASECO	Диалоговый процессор. Идентификация и исполнение основных команд оператора, связанных с обработкой программы и работой интерпретатора
BEDIDT	Таблицы ключевых слов интерпретатора, команд и констант
BASERR	Процессор обработки ошибок. Идентификация, вывод сообщений и передача управления в соответствующий модуль интерпретатора
BASMSG	Таблица сообщений об ошибках. Используется процессором обработки ошибок
BSFDSP	Ссылочная таблица и библиотека встроенных функций интерпретатора
BOPTFN	Модуль настройки путем корректировки объема секции

Структура и алгоритм работы интерпретатора

Интерпретирующая система Бейсик состоит из совокупности функциональных программных модулей и таблиц. Основными функциональными модулями являются: модули инициализации и подключения к операционной системе (STAP1), подготовки диалога с пользователем (READY), ввода и трансляции инструкций и операторов языка (EDIT), исполнения инструкций (COMDSP) и модуль интерпретации операторов языка (EXECUT).

Таблицы, входящие в состав системы Бейсик, можно разделить на три группы: синтаксические (времени трансляции), семантические (периода интерпретации) и рабочие,

Синтаксические таблицы используются модулем EDIT для распознавания операторов и инструкций и построения внутреннего кода программы. Эта группа представлена таблицами операций KEYWDS, ключевых слов KEYS и инструкций COMMND. Все эти таблицы имеют одинаковый формат вида

Текстовый ключ в ASCII	Код-заменитель
------------------------	----------------

Семантические таблицы используются модулями EXECUT и COMDSP, которые по своей структуре представляют собой набор таблицно управляемых семантических подпрограмм. Сами таблицы упорядочены по кодам-заменителям «адресные ссылки» на соответствующие подпрограммы: встроенных функций (TBSTAB), операторов языка (XADSP), инструкций управления интерпретатором (TABLE4) и функций, программируемых пользователем (FTABI).

В процессе работы интерпретатор создает и поддерживает ряд рабочих таблиц, основными из которых являются таблицы указателей строк языка LINTAB, символов SYMTAB и функций, определяемых пользователем DEFTAB.

Для общей координации работы интерпретатора строится еще одна таблица, называемая системной областью связи $\square\square$ OTSV. Эта таблица содержит указатели на другие таблицы интерпретатора, общие характеристики интерпретатора, описания логических каналов ввода-вывода и т. д.

Таким образом, занимаемая системой память делится на области программного кода интерпретатора, таблицы, области специального назначения и динамически распределяемую память для программ пользователя.

На рис. 1 представлена схема распределения памяти в Бейсике с учетом памяти, занимаемой операционной системой РАФОС. На рис. 2 показана общая структура интерпретирующей системы Бейсик, исходя из которой можно выделить следующие пути возможного расширения системы:

1. Расширение синтаксиса языка Бейсик включением в него новых понятий и объектов. Этот путь связан с включением в таблицы KEYS и XADSP новых синтаксических объектов и семантических подпрограмм и расширением модулей EDIT и EXECUT для соответствующей поддержки этих объектов. Данный способ позволяет органично включить в язык новое функциональное наполнение, однако он является наиболее сложным с точки зрения реализации.

2. Расширение посредством добавления к встроенным функциям языка новых функций. Это достигается включением в таблицы KEYS и

TBSTAB новых наименований функций и в модуль EXECUT подпрограмм, реализующих алгоритмы этих функций. Второй способ обладает меньшими синтаксическими возможностями, чем первый, по реализации его более проста, так как он не требует изменения синтаксического блока модуля EDIT.

3. Расширение возможностей Бейсика посредством функций, программируемых пользователем, — наиболее

<i>Страница ввода-вывода</i>
<i>РАФОС</i>
<i>Программно-моделируемый системой Бейсик стек</i>
<i>Область буферов системного назначения</i>
<i>Драйверы, буферные пулы, массивы, строковые переменные Бейсика</i>
<i>Временно-свободная память</i>
<i>Таблица SYMBOL</i>
<i>Таблица LINTAB</i>
<i>Таблица DEFTAB</i>
<i>Программа пользователя во внутреннем коде</i>
<i>Буфер входной строки с текстом</i>
<i>Системная область связи $\square\square$ OTSV</i>
<i>Программный код системы Бейсик</i>
<i>Аппаратный стек</i>
<i>Векторы прерываний</i>

Рис. 1

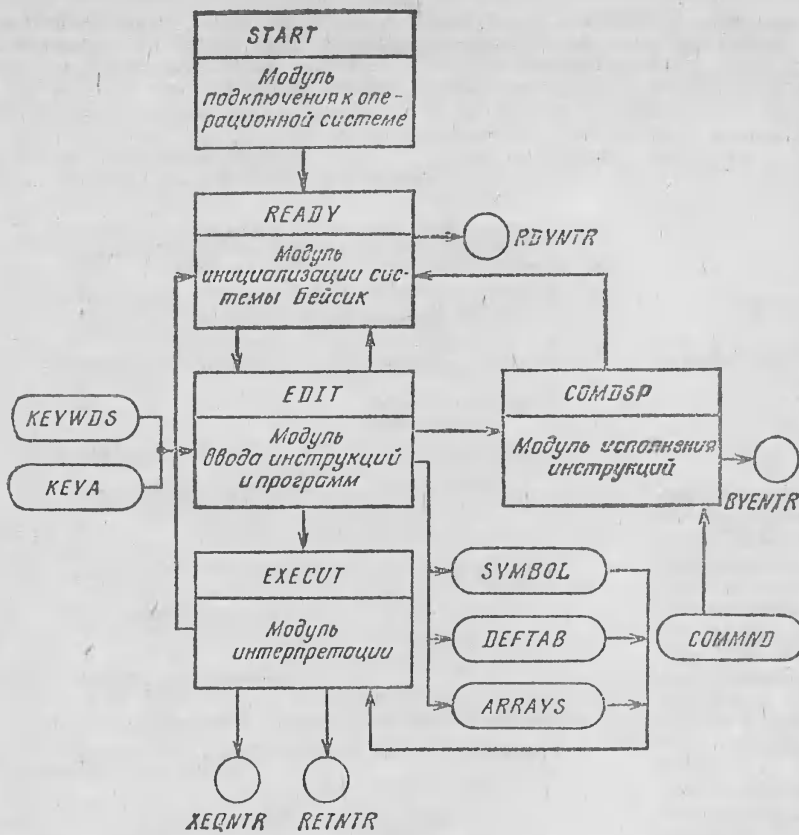


Рис. 2

Функции планирования

Наименование функции	Назначение	
	функции	параметров
FQUE (ID, LINE, TIME, FAZE, PRI)	Планирование процесса во времени	ID — идентификатор LINE — номер подпрограммы Бейсик TIME — период повторения FAZE — фаза запуска PRI — приоритет
DQUE (ID, LINE)	Отменить ранее спланированный процесс	ID — идентификатор LINE — номер подпрограммы Бейсик
FINT (VEC, LINE, PRI, CSR, MASK)	Планирование процесса по прерываниям	VEC — вектор устройства LINE — номер подпрограммы Бейсик PRI — приоритет CSR — регистр управления устройством
FINT (VEC, O)	Отменить ранее спланированный по прерываниям процесс	MASK — маска CSR VEC — вектор устройства

простой способ расширения Бейсика. Суть его заключается во включении в таблицу FTAB1 стандартного описания включаемой функции и программирования алгоритма функции. В этом случае система связывается через CALL-интерфейс. За простоту данного способа иногда приходится расплачиваться некоторой неэффективностью реализации, так как все функции CALL-интерфейса выполня-

ются только в период интерпретации. Однако на стадии отладки спецификаций и алгоритмов новых возможностей языка следует рекомендовать именно этот путь.

Вариант построения системы реального времени

Для программирования систем реального времени широко используется концепция параллельных вза-

модействующих процессов. Подобный механизм может быть легко реализован как расширение интерпретирующей системы Бейсик/РАФОС в виде пакета реального времени, что делает последний удобным инструментальным средством разработки систем управления технологическими процессами в реальном времени.

Разработка пакета реального времени для Бейсик/РАФОС базировалась на понятии процесса*, при этом для упрощения включения пакета в состав интерпретирующей системы синтаксис языка не подвергался расширению с целью ввода в него специальной конструкции «процесс». Концептуально процесс ассоциирован с понятием подпрограммы языка Бейсик, а механизм порождения потока управления вынесен на уровень пользователя в виде функций планирования процессов.

С точки зрения пользователя планирование представляет собой способ связывания внешних по отношению к вычислительному комплексу событий с определенной последовательностью вычислений по программе. В предлагаемой реализации допускается планирование по времени и планирование прерываний, что вполне достаточно для программирования большинства систем реального времени.

Связь пользователя с пакетом реального времени осуществляется при помощи функций внешнего уровня планирования FQUE, DQUE, FINT (см. таблицу), выполнение которых базируется на CALL-интерфейсе системы Бейсик/РАФОС. Для организации непосредственного управления дополнительными внешними устройствами в состав пакета введены специальные операции управления общей шиной.

Таким образом, наличие в системе Бейсик/РАФОС средств подключения дополнительных возможностей через CALL-интерфейс позволяет рассматривать процесс разработки пакета реального времени как проектирование независимого программного модуля, функциональная суть которого заключается в порождении и сопровождении нового по отношению к стандартному Бейсику потока управления. Программный модуль должен иметь точки сопряжения с интерпретатором Бейсик, что требует дополнительного перепрограммирования его реализации. В случае с Бейсик/РАФОС эту трудность легко преодолеть, так как разработчики системы ушли возможность подобного сопряжения посредством выделенных контрольных точек RETNTR, XEQNTR, RDYNTR, BYENTR (см. рис. 2). В базовой реализации Бейсик/РАФОС эти точки являются фиктивными,

* Янг С. Алгоритмические языки реального времени. Конструирование и разработка: Пер. с англ.— М.: Мир, 1985.

Выход интерпретатора на указанные точки сопряжения выполняет следующий интерфейс с пакетом реального времени: RDYNTR — инициализация, BYENTR — сброс, XEQNTR — запрос функции распределения интерпретатора между параллельными

процессами, RETNTR — сообщение об окончании очередного процесса. Выводы. Интерпретирующая система Бейсик/РАФОС допускает развитие ее функциональных возможностей и позволяет перестраивать не только набор поддерживаемых функций, но

и языковые структуры языка. Это позволяет доступными средствами реализовать специализированную Бейсик-подобную систему. Телефон для справок: 911-396, Саратов. Статья поступила 16 декабря 1986 г.

УДК 681.3.06

С. С. Селицкий, М. Ю. Сыркин

ПРОЦЕДУРА ПЕРЕМЕЩЕНИЯ ЧАСТЕЙ ЗАГРУЗОЧНОГО МОДУЛЯ ДЛЯ МИКРОПРОЦЕССОРА КР580ИК80

В процессе комплексной отладки программного обеспечения встроенных МП-систем обычно используют мониторные программы, содержащие минимальный набор необходимых процедур, таких как просмотр и модификация содержимого памяти и регистров, задание точек останова и др. При этом обнаружение любой ошибки требует обращения к инструментальным средствам для внесения исправления и трансляции, после чего программа вновь вводится в систему и вновь тестируется.

Стремясь сократить число обращений к инструментальным средствам в процессе отладки, обнаруженные ошибки нередко накапливают с тем, чтобы за один раз исправить максимальное их число. Для этого ошибки «блокируют» с помощью дополнительных команд перехода («заплат»). Этот распространенный прием запутывает программу, затрудняет ее чтение и часто приводит к новым ошибкам.

Для отладочного монитора предлагается процедура TAILOR (рис. 1), позволяющая:

перемещать участки загрузочного модуля программы с соответствующей коррекцией адресных команд перехода; вставлять вместо «заплат» недостающие команды (при этом область программы, начиная с места вставки, смещается на соответствующее число байтов в сторону старших адресов); организовать просмотр всей программы, выделение адресных команд перехода и коррекцию адресных частей тех команд, которые ссылаются на перемещаемую область программы; удалять «лишние команды» смещением части программы; перенастраивать отлаженный загрузочный модуль на любую область адресного пространства встроенной МП-системы после исправления всех ошибок непосредственно в отладочном ОЗУ.

Описание процедуры. Процедура оперирует с программой пользователя, загруженной в отладочное ОЗУ встроенной МП-системы. Ее работа осуществляется в два этапа (два прохода по памяти команд).

На первом этапе выделяются адресные команды передачи управления (условный и безусловный переходы, условный и безусловный вызовы подпрограммы) и производится проверка попадания их адресной части (ADR) в перемещаемую область программы (BEGSHIFT...ENDSHIFT). Если фиксируется попадание, т. е. $BEGSHIFT \leq ADR \leq ENDSHIFT$, то к ADR прибавляется фиксированная величина сдвига фрагмента SHIFT, после чего она записывается на прежнее место — в поле адреса обрабатываемой команды перехода.

Второй этап — пересылка перемещаемого фрагмента программы на новое место в памяти. При этом предусмотрена возможность задавать режимы обнуления и сохранения области источника. Первый режим используется для вставки и удаления команд, второй — для перепайки всей программы пользователя на

```

; ПРОЦЕДУРА ВСТРОЕННОГО МОНИТОРА, ПЕРЕМЕЩАЮЩАЯ ЧАСТЬ ПРОГРАММЫ
; ПОЛЬЗОВАТЕЛЯ В СТОРОНУ СТАРШИХ ИЛИ МЛАДШИХ АДРЕСОВ С СООТВЕТ-
; СТВУЮЩЕЙ КОРРЕКЦИЕЙ АДРЕСНЫХ КОМАНД ПЕРЕХОДА
;
; ВХОДНЫЕ: BEGPRO - АДРЕСА НАЧАЛА И КОНЦА ПРОСМАТРИВАЕМОЙ
; ПАРАМЕТРЫ: ENDPRO - ОБЛАСТИ ПОЛЬЗОВАТЕЛЬСКОЙ ПРОГРАММЫ
; BEGSHIFT - АДРЕСА НАЧАЛА И КОНЦА ПЕРЕМЕЩЕНОГО
; ENDSHIFT - СЕКЦИОНА ПРОГРАММЫ
; SHIFT - ВЕЛИЧИНА СМЕЩЕНИЯ СЕКЦИОНА ( С ОЗНАКОМ )
; CSN - ЯЧЕЙКА РЕЖИМА РАБОТЫ ПРОЦЕДУРЫ
;
; ОБРАЩЕНИЕ К ПРОЦЕДУРЕ:
; CALL TAILOR
;
; ИНИЦИАЛИЗАЦИЯ

```

```

TAILOR:: LXI H,0 ; СОХРАНИТЬ УКАЗАТЕЛЬ
DAD SP ; СТЕКА ВЫЗЫВАТЕЛЯ
SHLD 010STACK ; ПРОГРАММЫ
LXI SP,STACK

```

; ЗАГРУЗКА МАКРАЛЬНЫХ ДАННЫХ СОБРАТЧИВЫХ ПРОГРАММ

```

LHLD SHIFT ;
PUSH H ; BC = ВЕЛИЧИНА СДВИГА
POP B ;
;
LHLD ENDPRO ; DE = АДРЕС КОНЦА ПРОГРАММЫ
XCHG ;
INX D ;
;
LHLD BEGPRO ; HL = АДРЕС НАЧАЛА ПРОГРАММЫ

```

; ВЫДЕЛЕНИЕ ТРЕХБАЙТНЫХ, ДВУХБАЙТНЫХ КОМАНД И КОМАНД ПЕРЕХОДА

```

NEXT:: MOV A,M
CPI CALL ; ПРОВЕРКА НА "CALL"
JZ CHECK ;
CPI JMP ; ПРОВЕРКА НА "JMP"
JZ CHECK ;
ANI 11110111B ; ПРОВЕРКА НА "IN", "OUT"
CPI 11010011B
JZ L2 ;
ANI 11100111B ; ПРОВЕРКА НА "LHLD", "SHLD", "LDA", "STA"
CPI 00100010B
JZ L3 ;
ANI 11000111B ; ПРОВЕРКА НА "MVI"
CPI 00000110B
JZ L2 ;
CPI 11000110B ; ПРОВЕРКА НА "ADI" И Т.Д.
JZ L2 ;
CPI 11000010B ; ПРОВЕРКА НА УСЛОВИЯ "JUMP"
JZ CHECK ;
CPI 11000100B ; ПРОВЕРКА НА УСЛОВИЯ "CALL"
JZ CHECK ;
MOV A,M
ANI 11001111B
CPI 00000011B ; ПРОВЕРКА НА "LXI"
JZ L3 ;
JMP L1 ; ДЛЯ ОДНОБАЙТНЫХ КОМАНД

```

; ПЕРЕХОД К СЛЕДУЮЩЕЙ КОМАНДЕ

```

SKIP:: POP Y
POP Z ; ДЛИНА КОМАНДЫ:
; 3 БАЙТА
; 2 БАЙТА
; 1 БАЙТ
;
CALL COMP ; ПРОВЕРКА НА КОНЕЧНЫЙ АДРЕС
JNC NEXT ;

```

; ПЕРЕМЕЩЕНИЕ

```

MOVE:: MVI A,RET ; ЗАНЕСТИ "RET" В ОБЛАСТЬ КОМАНД
STA COM+3 ;
;
LXI B,0101BN ; ЗАНЕСТИ КОДЫ КОМАНД "DCH B" И "DCH 0"
;
LHLD SHIFT ; ОПРЕДЕЛИТЬ ЗНАК СДВИГА
DAD H ;
;
LHLD ENDSHIFT ;
XCHG ; ЗАГРУЗИТЬ ГРАНИЦ ОБЛАСТИ СДВИГА
LHLD BEGSHIFT ;
;
JNC FORWARD

```

```

BACKWARD: LXI B,0313H ; ЕСЛИ СДВИГ В СТОРОНУ МЛАДШИХ АДРЕСОВ
          XCHG ; ЗАМЕНИТЬ НА КОДЫ КОМАНД "INX H" И "INX D"
          ; ПОНЯТЬ МЕСТАМИ ENDSHIFT И BEGSHIFT

FORWARD:  PUSH B ; ИНИЦИАЛИЗАЦИЯ
          XTHL ; ОБЛАСТИ
          SHLD COM+1 ; КОМАНДА

          LHL D,SHIFT ; ИНДЕКСАЦИЯ НА ВЕЛИЧИНУ СДВИГА
          DAD D ;
          XTHL ;
          POP B ;

CYCLE:    CALL COMP ; ЦИКЛ ПЕРЕСЫЛКИ
          LDAX D ;
          STAX B ;
          MVI A,0 ; ОБНУЛЕНИЕ ОБЛАСТИ-ИСТОЧНИКА
          ; (ЕСЛИ ТРЕБУЕТСЯ)
          CALL COM ; ПЕРЕДАЧА УПРАВЛЕНИЯ В ОБЛАСТЬ КОМАНД
          JNZ CYCLE ;

          LHL OLDSTACK ; ВЕРНУТЬ ПРЕЖНИЙ УКАЗАТЕЛЬ СТЕКА
          SPHL ;
          RET ; *** КОНЕЦ ***

```

```

; ДЛЯ КОМАНД ПЕРЕХОДА : ПРОВЕРКА НА ПОДАВЛЕНИЕ АДРЕСА В
; ; ПЕРЕДАЧЕМОЙ ОБЛАСТИ

```

```

CHECK:    PUSH D
          PUSH H

          INX H ;
          MOV E,H ; ЗАГРУЗИТЬ АДРЕСНУЮ ЧАСТЬ КОМАНДЫ
          INX H ;
          MOV D,H ;

          LHL BEGSHIFT ; СРАВНИТЬ С АДРЕСОМ НАЧАЛА СЕГМЕНТА
          CALL COMP ;
          JC SKIP ;

          LHL ENDSHIFT ; СРАВНИТЬ С АДРЕСОМ КОНЦА СЕГМЕНТА
          INX H ;
          CALL COMP ;
          JNC SKIP ;

```

МОДИФИКАЦИЯ АДРЕСА ПЕРЕХОДА

```

          XCHG ;
          DAD B ; ПРИБАВИТЬ ВЕЛИЧИНУ СДВИГА
          XCHG ;

          POP H ;
          INX H ;
          MOV H,E ;
          INX H ; ВЫГРУЗИТЬ АДРЕСНУЮ ЧАСТЬ КОМАНДЫ
          MOV H,D ;
          POP D ;

          JMP L1 ; ПЕРЕХОД К СЛЕДУЮЩЕЙ КОМАНДЕ

```

```

; СРАВНЕНИЕ DE И HL :
; CY => DE < HL
; Z => DE = HL

```

```

COMP:     MOV A,D ; СРАВНЕНИЕ СТАРШИХ БАЙТОВ
          CMP H ;
          RNZ ;
          MOV A,E ;
          CMP L ; СРАВНЕНИЕ МЛАДШИХ БАЙТОВ
          RET ;

```

ОБЛАСТЬ ДАННЫХ
; ЗАПОЛНЯЕТСЯ ПОЛЬЗОВАТЕЛЕМ ПЕРЕД ВЫЗОВОМ :

```

BEGPRO:   DS 2 ; НАЧАЛО И КОНЕЦ
          DS 2 ; ПРОСМАТРИВАЕМОЙ ОБЛАСТИ
ENDPRO:   DS 2 ; НАЧАЛО И КОНЕЦ
          DS 2 ; ПЕРЕМАЩАЕМОЙ ОБЛАСТИ
ENDSHIFT: DS 2 ; ВЕЛИЧИНА СДВИГА
          DS 2 ;
SHIFT:    DS 2 ;
          DS 1 ; ЯЧЕЙКА РЕЖИМА РАБОТЫ
          DS 1 ;

```

ИСПОЛЬЗУЕТСЯ ПРОЦЕДУРОЙ ПРИ РАБОТЕ :

```

OLDSTACK: DS 3 ; ОБЛАСТЬ КОМАНД
          DS 2 ; УКАЗАТЕЛЬ СТЕКА
          DS 2 ; ВЫЗЫВАЮЩЕЙ ПРОГРАММЫ
          DS 6 ;
          EQU X ; СТЕК ПРОЦЕДУРЫ

```

END TAILOR

Рис. 1. Процедура TAILOR для отладочного монитора

другую область адресного пространства. Пересылка начинается с младших или старших адресов памяти в зависимости от направления перемещения фрагмента программы.

Применение процедуры. Пусть требуется вставить в середину оттранслированной и загруженной программы, начиная с адреса D, дополнительный фрагмент

(рис. 2). Для этого необходимо «раздвинуть» две части программы, переместив, например, вторую часть в сторону старших адресов и соответственно скорректировав имеющиеся в программе ссылки.

A:	...	0000:	...
B:	JMP C	0100:	C3 0002
C:	JMP E	0200:	C3 0004
D:	...	0300:	...
E:	JMP F	0400:	C3 0005
F:	JMP B	0500:	C3 0001
G:	...	0600:	...
A)		B)	

Рис. 2. Отлаживаемая программа до обращения к процедуре TAILOR:

а) фрагмент программы на ассемблере;
б) содержание памяти

Прежде чем воспользоваться процедурой TAILOR, требуется занести в область данных ОЗУ значения входных параметров (см. таблицу). Если требуется сразу очистить область для вставляемого фрагмента, то необходимо задать режим обнуления области-источника, записав по адресу COM блока данных код команды STAX D (12H). В противном случае по этому адресу заносится код команды NOP (00H).

Параметр	Символическое имя	Что заносится	Числовое значение
Начало просматриваемой области	BEGPRO	A	00H
Конец просматриваемой области	ENDPRO	G	600H
Начало перемещаемой области	BEGSHIFT	D	300H
Конец перемещаемой области	ENDSHIFT	G	600H
Величина сдвига	SHIFT	K	10H
Режим работы	COM	NOP или STAX D	00H или 12H

После вызова процедуры TAILOR отлаживаемая программа примет вид, показанный на рис. 3. Участок программы, начиная с метки D, переместился на величину сдвига в сторону старших адресов. Адресные команды перехода в пределах всей программы скорректированы так, что все ссылки на перемещаемый фрагмент модифицированы на величину сдвига, а ссылки на неподвижный фрагмент остались прежними. Таким образом, все команды передачи управления в отлаживаемой программе останутся корректными. Область,

A:	...	0000:	...
B:	JMP C	0100:	C3 0002
C:	JMP E+K	0200:	C3 1004
D:	NOP	0300:	00
E:	NOP
F+K:	JMP F+K	0410:	C3 1005
G+K:	JMP B	0510:	C3 0001
H:	NOP	0610:	00
A)		B)	

Рис. 3. Отлаживаемая программа после обращения к процедуре TAILOR:

а) фрагмент программы на ассемблере;
б) содержание памяти

предназначенная для вставки, заполнена нулями (код команды NOP).

Применение процедуры для удаления из программы «лишних» команд отличается тем, что параметр SHIFT задается отрицательным в дополнительном коде. При этом сдвиг фрагмента происходит в сторону младших адресов, «лишние» команды удаляются, ссылки на перемещаемую область корректируются. Освободившиеся в конце программы ячейки памяти заполняются нулями.

Для настройки отлаженного программного модуля на определенную область адресного пространства следует задать $BEGPRO=BEGSHIFT+ENDPRO=ENDSHIFT$ (область сдвига совпадает с просматриваемой областью). В ячейку с адресом COM следует записать код команды NOP, задав тем самым режим сохранения содержимого области-источника. Если в процессе перемещения программы запись в область-получатель физически невыполнима, то можно воспользоваться копией настроенной программы, сохранившейся в области-источнике. Затем программа должна быть перенесена в область исполнительных адресов имеющимися для этого в системе средствами (например, запись в ИЗУ).

Условия.

1. BEGPRO должен быть адресом команды.
2. Просматриваемая область (BEGPRO..ENDPRO) не должна содержать данных или сторонних кодов, которые могут сбить программный счетчик при выделении команд перехода. Свободные фрагменты памяти внутри просматриваемой области лучше всего заполнить кодом команды NOP (00H).

УДК 681.3

А. В. Воржев, Б. С. Зверков, А. И. Кикоть, Е. В. Яковлева

ИНТЕГРИРОВАННАЯ СИСТЕМА РАЗРАБОТКИ РАСШИРЯЕМЫХ КРОСС-СИСТЕМ НА БАЗЕ ДВК-2М ДЛЯ МИКРОПРОЦЕССОРОВ

Введение.

Освоить новые микропроцессорные комплекты (МПК) и успешно их применить при проектировании систем обработки информации часто трудно из-за недостатка средств проектирования программного обеспечения

(ПО) микропроцессорных систем (МПС) и их отладки.

Но так ли необходимы для разработки программного обеспечения готовые средства?

Как показывает опыт, при наличии ДВК-2М с операционной системой

3. В программе не должно быть внешних по отношению к просматриваемой области памяти ссылок на перемещаемый фрагмент. В противном случае эти ссылки не будут скорректированы процедурой TAILOR, что сделает невозможной дальнейшую отладку программы.

4. Поскольку адресная часть команд обработки данных (LDA, LXI и др.) не корректируется процедурой TAILOR, ее нельзя применять для программ, использующих эти команды для организации переходов, например

```

..... или .....
LXI H, ADDR      LXI H, ADR
PCHL              PUSH H
                  RET
    
```

В таких фрагментах программы за коррекцией адреса перехода приходится следить программисту.

Предлагаемая версия процедуры используется совместно с монитором универсального контроллера «Электроника К1-20», однако она может применяться и с любым другим монитором, работающим в указанной системе координат*.

* Из рецензии. Предложенная подпрограмма была опробована на микроЭВМ «Микроша» и оказалась весьма удобной и полезной.

Телефон для справок: 335-90-62, Москва. Селицкий Семен Станиславович.

Статья поступила 21 октября 1986 г.

ОС ДВК, в которую включены стандартные программы МАКРО-11, LINK, LIBR, CREF, ODT, а также некоторые средства программного комплекса РТК-микро, знакомясь с документацией, описывающей МПК, можно создавать инкрементную (последовательно наращиваемую) базу знаний, позволяющую проектировать и отлаживать ПО для создаваемых МПС.

Алгоритм построения такой базы знаний (БЗ) приведен на рис. 1 и не содержит никаких необычных процедур для человека, знакомого с работой ОС ДВК и языком МАКРО-11.

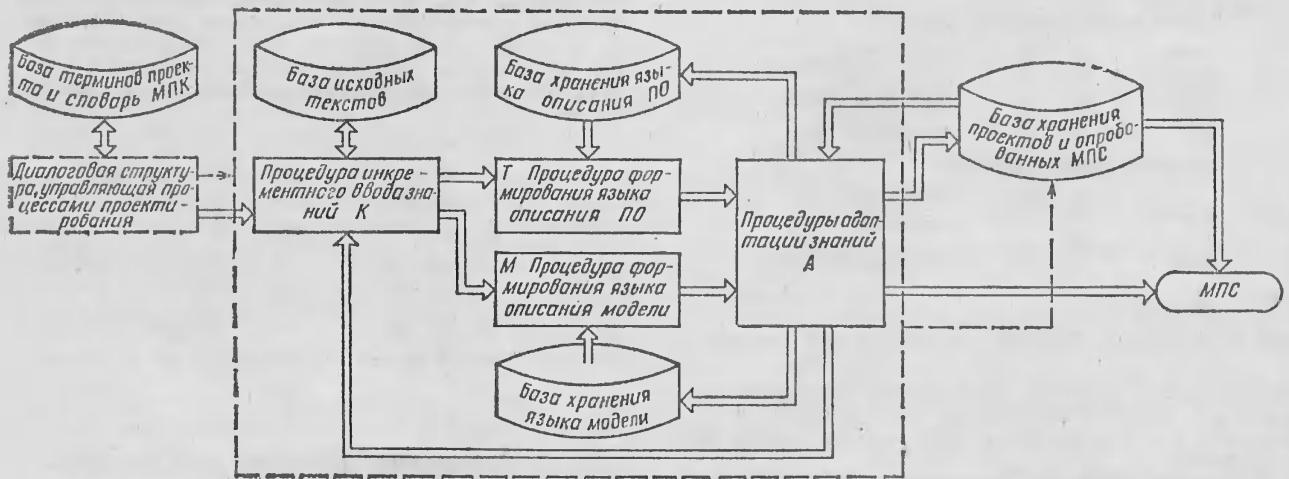


Рис. 1 База знаний для проектирования ПО МПС

БЗ строится с помощью ввода информации о функционировании МПК (назовем этот процесс «конспектированием») (К) на языке МАКРО-11, который в данном случае используется в качестве текстового процессора).

Цель конспектирования — создание в ОС ДВК:

— процедур имитационного моделирования МПК на основе той информации, которая дана в справочной литературе;

— транслятора с языка программирования МПК (совпадающего по синтаксису с МАКРО-11) в коды, которые будут использоваться сначала имитационной моделью при отладке ПО, а затем реальной МПС;

— транслятора с языка описания модулей различных конфигураций МПС, создаваемых на базе МПК и опыта разработчиков.

Такая целевая установка приводит нас к разделению обработки потока информации, создаваемой в процедуре (К), на две части: формирование языка описания ПО или генерации ПО (Г) и языка описания модели или моделирования (М). «Конспектирование» ведется таким образом, что знания о процессе кодирования и управлении ресурсами

МПК отделяются от знаний о функционировании отдельных частей МПК.

Знания о процессе кодирования и управлении ресурсами МПК помещаются в базу хранения «языка описания ПО» и представляют собой тексты макроопределений тех понятий, которые используются в документации на МПК. После наполнения этой части БЗ можно, используя ее и МАКРО-11, обрабатывать тексты описания алгоритмов в понятиях, выбранных для данного МПК. В результате такой обработки текстов генерируется программа в объектном коде для загрузки в МПС. Одновременно со знаниями о кодировании в описании МПК встречаются знания о функционировании отдельных частей МПК.

Процессам, протекающим в МПК, можно поставить в соответствие функционально аналогичные процессы, протекающие в ДВК, так как и МПК и ДВК относятся к классу фон-неймановских вычислительных систем. Макроопределения алгоритмов функционирования МПК пишутся на мнемокоде ДВК и составляют базу хранения языка описания модели МПК. Используя эту часть БЗ и

МАКРО-11, можно, обработав текст описания МПС, сгенерировать программу для ДВК, моделирующую работу МПК. Эта программа в процессе исполнения использует объектный код, получаемый при генерации программы для МПС базой хранения «языка описаний ПО», производит все те действия, которые производила бы реальная МПС, собранная в соответствии с текстом описания.

Таким образом, при изучении МПС по документации, затратив некоторые усилия на осмысление, структуризацию получаемой информации и внесение ее в БЗ на ДВК-2М, можно получить кросс-систему, реально пригодную для работы по проектированию ПО МПС.

Отметим, что среди указанных свойств кросс-системы наиболее важное — возможность вносить в нее дополнительные знания о технологии программирования, облегчающие работу по проектированию ПО.

Построение кросс-транслятора в виде базы знаний о микропроцессоре.

Рассмотрим более подробно технологию создания базы знаний, предназначенной для проектирования ПО МПС, на примере МПК КР580, (БЗ названа «РАСА-84»).

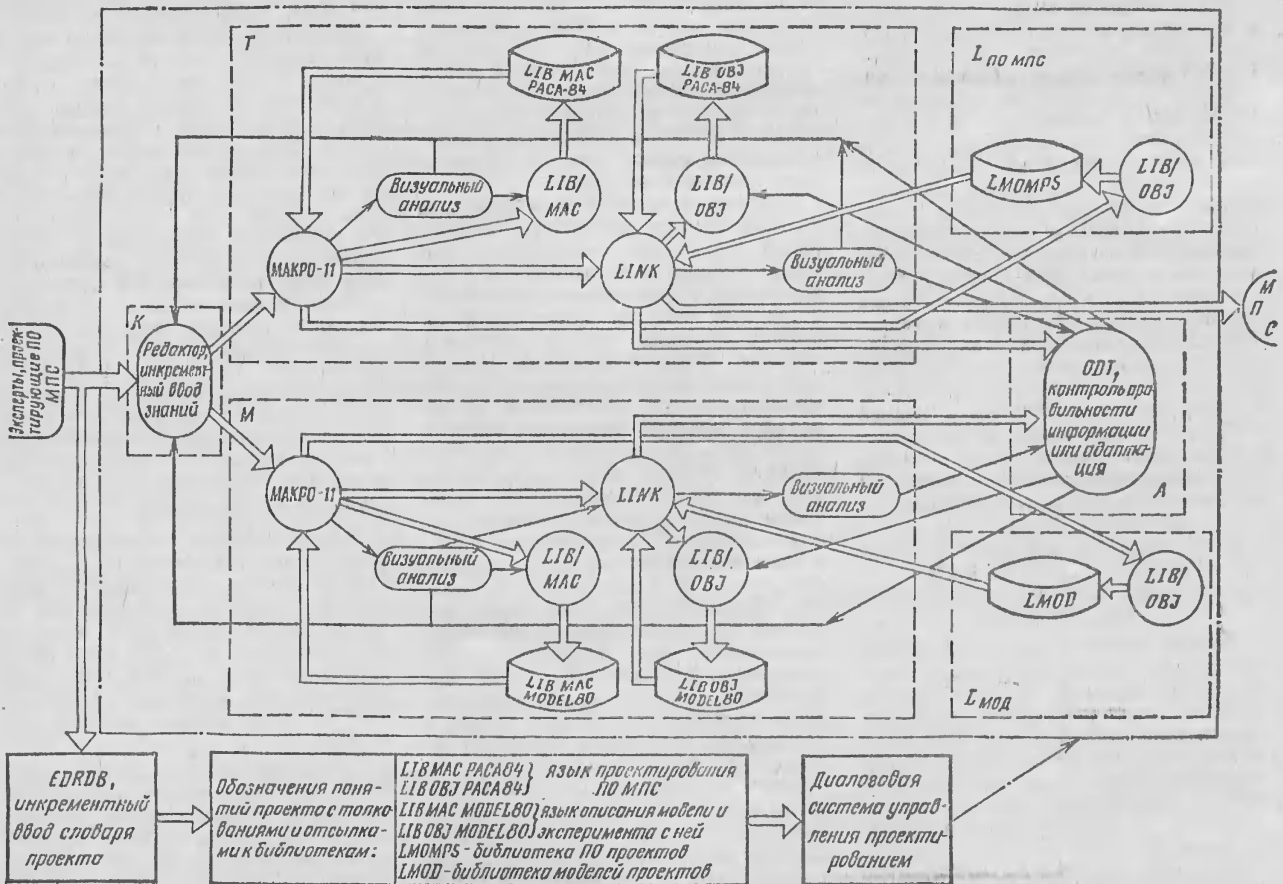


Рис. 2. Цикл ввода и адаптации информации

Первичные знания, формировавшие «PACA-84», — описание команд KP580ИК80 и интерфейсных БИС.

Ввод описаний по данным [1] выполнен на языке МАКРО-11 с помощью одного из экранных редакторов. Как уже было сказано, весь цикл ввода и адаптации информации (рис. 2) состоит из двух частей: ввода текстов макроопределений понятий для процесса генерации ПО для KP580 (Г); ввода текстов макроопределений функционирования элементов МПК KP580 (М).

Отметим, что часть процесса (Г), а именно «конспектирование» знаний о кодировании, управлении ресурсами на уровне системы команд процессора и о распределении памяти, а также все ограничения, накладываемые на него в ДВК, хорошо описана [2].

Скажем только, что (Г) представляет собой описание на МАКРО-11 мнемоники команд KP580ИК80 в виде макроопределений, которые будут расширяться в процессе трансляции в объектный код этого процессора. Знания о кодировании и командах процессора — первичные и составляют основы базы хранения «языка описаний ПО». Тексты макроопределений вводятся текстовым редактором [1].

После ввода проверяется правильность задания информации прогноза текста через текстовый процессор МАКРО-11. Если он выдает правильный листинг преобразования мнемоники в объектный код (это проверяется визуально), то текстовый модуль считается прошедшим адаптацию. В этом случае макроопределение можно поместить (программой LIBR) в макробблиотеку мнемоник процессора KP580ИК80 (первый раздел LIBMAC «PACA 84»).

Одновременно с заполнением этого раздела можно приступить к созданию других разделов. Так, раздел структур управления можно реализовать на основании знаний о структурном программировании и знаний о передаче управления в микропроцессор. Этот раздел будет состоять из макроопределений операторов структурного программирования:

```
IF...THEN...ELSE...END
WHILE...DO...END
REPEAT...UNTIL...END
SELECT OF...DO...
OF...DO...END
```

Другие разделы, например раздел организации выполнения процессов в реальном масштабе времени, формируются после завершения «конспектирования» знаний о коде ЦП МПК и являются описанием знаний об управлении процессами, которые имеются у разработчика или которые он хотел бы опробовать, взяв из литературы [3].

Описание знаний типа «управление процессами» разделяется на две части: макроопределения структур дан-

ных и процедур работы с ними, исполнительные модули, реализующие стандартные алгоритмы процедур.

Макроопределения после визуального контроля на адаптацию помещаются в соответствующий раздел LIBMAC «PACA-84». Исполнительные модули транслируются с визуальным контролем, после чего помещаются в соответствующий раздел ЛИБОВЛ «PACA-84». Затем визуально контролируется адаптация объектного модуля: транслируется макровызов соответствующей процедуре управления и полученный объектный код обрабатывается редактором связей LINK, ко входу которого подключена ЛИБОВЛ. Визуально проверяется карта связей, полученная в результате обработки LINK. Так, обработанный модуль еще не считается полностью адаптированным, поскольку процедура, его реализующая, не прошла отладку в МПС. На этом пункте мы прервем описание процесса заполнения (Г) и вернемся к нему после обсуждения процесса заполнения (М).

Построение модели МПК в виде базы знаний о функционировании МПК.

Осуществляя процесс (Г), мы можем одновременно работать над процессом (М), заполняя вторую часть нашей базы знаний — базу хранения «языка описания модели».

Как и в случае (Г), содержимое «базы хранения языка модели» — это макроопределения, содержание знания о функциях, которые может выполнить KP580, и алгоритмы, описывающие эмуляцию этих функций в виде программ на ДВК.

Описание функций в виде программ на ДВК требует дополнительных операций (таких, как отладка), которые должны быть проведены с информацией о функционировании МПК перед тем, как она будет помещена в базу хранения «языка описания модели». После визуального контроля трансляции макровызова модель микропроцессора описывается на основании понятий, введенных ранее с включением новой информации. Как правило, по мере ввода информации и ее адаптации формируются заготовки моделей, эмулирующих с разной степенью подробности МПК, в виде текстов их описаний, которые содержат уже адаптированные знания. Следовательно, известно, что модель без подключения к ней новой информации уже функционировала правильно. Подключение новой информации к модели может отразиться на ее работоспособности только в том случае, если новая процедура не выполняет своих функций или выполняет их неверно. Для выявления ошибок в новой информации сгенерированную и собранную модель процессора вместе с информацией, необходимой для ее функционирования, помещают в ДВК и отлаживают новую функцию.

На этом этапе знания адаптируются на основе проверки совместного функционирования частей (Г) и (М) базы знаний: отладочные примеры описываются в терминах знаний, вложенных в (Г), а модель — в терминах знаний, вложенных в (М). После этого проводятся трансляции описаний и общая сборка. Полученный таким образом загрузочный модуль прогоняется в ДВК, и его работоспособность оценивается по результатам прогона с ODT.

Отметим, что ошибки обнаруженные при прогонах, заставляют более тщательно изучить описание процессора и часто выявляют отсутствие в описании той или иной информации, необходимой для его правильного использования.

После наполнения базы знаний до определенного уровня, происходит интегрирование знаний. Используя совместно обе части базы, вы можете описать программы для применения микропроцессора и свойства той его модели, на которой будете отлаживать эти программы. Каждый разработчик сам определяет момент, когда он имеет необходимый и достаточный для его целей язык описания модели и язык написания программ. В этом момент дальнейшее наполнение базы знаний можно прекратить.

Однако из опыта известно, что по мере работы с новым объектом вы узнаете о нем все больше нового, и этот опыт вы хотели бы использовать в дальнейшем. Повторяя циклы (Г) и (М), можно легко накапливать опыт, вводя новые команды языка программирования и новые свойства модели. Так, при заполнении БЗ «PACA-84» операторами управления процессами мы использовали свой опыт написания ОС реального времени и создали для мультипроцессорной системы ряд операторов (системных макросов) управления процессами реального времени и мультипрограммирования, а также ряд операторов описания и управления вводом-выводом. Все эти операции были отлажены на статической модели, т. е. без учета временных характеристик процессора.

Это, естественно, не исключает наличия в них динамических ошибок. Однако знание о возможности динамических ошибок позволяет при переходе на реальную МПС или модель, содержащую знание о временных характеристиках KP580ИК80, сосредоточить свое внимание на ситуациях, которые могут выявить эти ошибки.

Отладочная стратегия строится так, чтобы наиболее полно выявить все временные несоответствия, возможные в ПО. Таким образом, кросс-система, построенная по принципу БЗ, может локализовать точки возможных ошибок (за счет прогнозирования их).

«ЭЛЕКТРОНИКА БК-0010» В СИСТЕМАХ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ОБЪЕКТАМИ

УДК 681.322.1—181.4

П. В. Полянский, Н. А. Ширковский

Отечественная промышленность выпускает для разработки производственных систем автоматизации управляющие вычислительные комплексы (УВК) и программируемые контроллеры (ПК).

Применение для автоматизации отдельных технологических установок традиционных УВК, построенных, в основном, на базе мини-ЭВМ, неэффективно из-за избыточности, сложности интерфейсных устройств и высокой стоимости.

Автономные ПК позволяют реализовать лишь простейшие функции линейных регуляторов или программаторов процесса по времени, а встраиваемые ПК, как правило, не обеспечены требуемой для их эффективного использования инфраструктурой. Это практически исключает возможность их самостоятельного применения потребителем.

Одноплатные персональные ЭВМ (ПЭВМ) «Электроника БК-0010», соединяющие свойства универсальных ЭВМ и минимальных контроллеров, позволяют заполнить экологическую нишу между УВК и ПК. Эти простейшие ПЭВМ выгодно отличаются от контроллеров наличием встроеной универсальной клавиатуры, видео-контроллера, средств связи с накопителем на магнитной ленте, повышенным объемом ОЗУ (32К байт). «Электроника БК-0010» программируется на языке высокого уровня с графическими функциями. Ввод данных — интерактивный, без каких-либо дополнительных инструментальных средств. Компактность, повышенная надежность и низкая стоимость компьютеров этого типа определяют перспективность их применения в системах управления.

С шестатыными устройствами ПЭВМ связана через два ее разъема. На первый выведена внутренняя магистраль машины (типа Q-bus), на второй — программируемый параллельный порт ввода-вывода и линия прерывания ПРТ для организации таймера-счетчика событий.

При стандартном подключении устройства связи с объектом (УСО) к ПЭВМ (через магистраль) возникают существенные трудности по ее надежной буферизации и дешифрации адреса внешнего устройства. Эффективный обмен с УСО возможен через программируемый параллельный порт ввода-вывода, состоящий из двух 16-разрядных регистров. Обращение к ним происходит по одному адресу, но один регистр доступен только для чтения (будем называть его портом ввода), а другой только для записи (порт вывода). Оба порта реализованы на многорежимных буферных регистрах (МБР) К589ИР12 [1].

Для организации обмена порт вывода ПЭВМ разбивается на две зоны; младшие разряды — для выдачи данных, старшие разряды — для формирования адреса внешнего устройства и управляющих сигналов. Максимальная длина зоны данных обычно определяется разрядностью ЦАП в УСО.

Пример минимальной аппаратной реализации — УСО для управления медленнореагирующей технологической установкой. Оно обеспечивает ввод восьми, вывод одного аналогового сигнала и подсчет текущего времени. Схема устройства (рис. 1) классическая [2]: входные аналоговые сигналы через диодную защиту поступают на входы интегрального аналогового мультимплексора (К561КП2), далее — на буферный повторитель (К140УД7), с выхода которого на вход интегрального АЦП (К1113ПВ1). Цифровой код с выхода АЦП

подается непосредственно на порт ввода ПЭВМ (разряды 0...9). Выходные ТТЛ-уровни порта вывода ПЭВМ с КМОП-уровнями адресных входов аналогового мультимплексора согласует микросхема с открытым коллектором К155ЛА8.

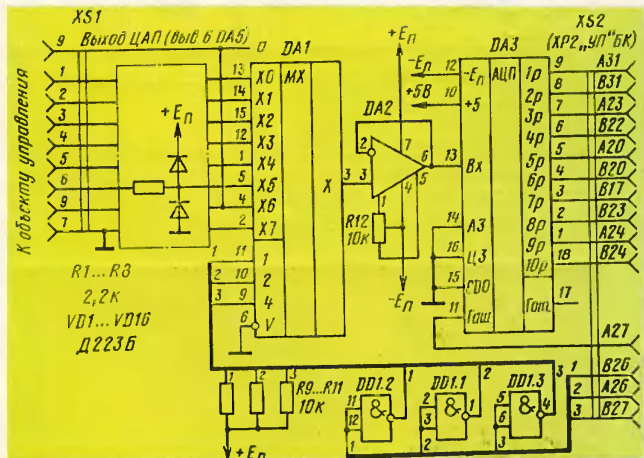


Рис. 1. Принципиальная схема АЦП минимального УСО

К разрядам 0...9 порта вывода (рис. 2) подключены цифровые входы ЦАП (К572ПА1А и работающая с ней в паре К140УД7). Разряд 10 порта используется для подачи сигнала «Гашение АЦП», разряды 11...13 — для адресации канала мультимплексора.

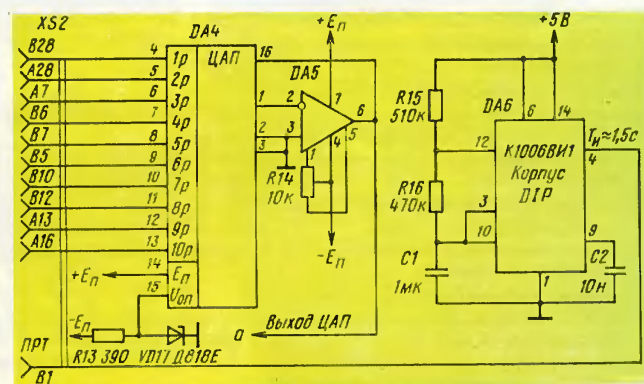


Рис. 2. Принципиальная схема ЦАП и таймера минимального УСО

Таймер (см. рис. 2), аппаратно поддерживающий функцию FCLK() в языке Фокал, выполнен на ИМС К1006В1. Сигнал с ее выхода подан на линию ПРТ.

Благодаря малому количеству микросхем энергопотребление низкое. Поэтому УСО и компьютер могут получать питание от одного источника. Для двуполярного питания аналоговых микросхем применен высокочастотный стабилизированный преобразователь [3], дополненный «мягким запуском» и параметрическим стабилизатором выходного напряжения (рис. 3). Общий ток, потребляемый УСО, не превышает 300 мА. Устройство (рис. 4) с преобразователем питания размещается в корпусе 55×55×145 мм.

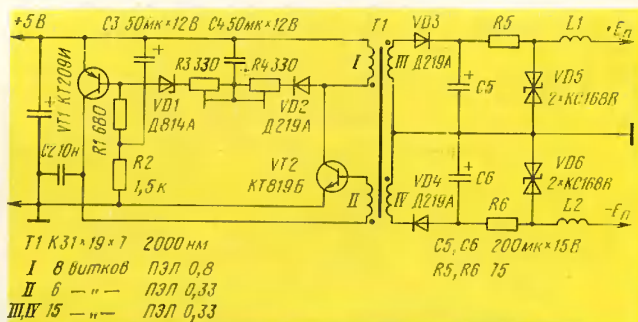


Рис. 3. Принципиальная схема преобразователя питания минимального УСО

Программирование ПЭВМ на языке высокого уровня позволяет легко описывать качество управляемого процесса (сложной функции состояния), использовать тривиальные алгоритмы управления.



Рис. 4. Внешний вид минимального УСО для «Электроники БК-0010»

«Электроника БК-0010» в комплексе с простейшим УСО была использована в системе управления процессом высокотемпературной пайки крупногабаритных изделий в индукционных печах.

Алгоритм работы системы управления обеспечивает сбор, первичную обработку, индикацию и сохранение на магнитной ленте данных от трех терморезисторов и текущего времени; индикацию пестчатых режимов; автоматическое управление режимом нагрева.

Первичная обработка вводимых данных — это программная компенсация дрейфа нуля и калибровка цепи аналого-цифрового преобразования, трехточечное усреднение результатов измерений.

В ходе процесса на экране монитора показываются временные графики и числовые значения измеряемых температур.

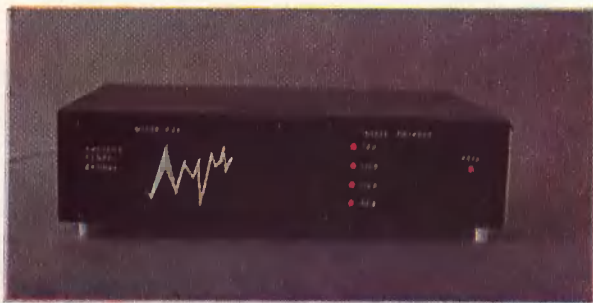


Рис. 5. Система сбора данных о распределении энергии в электронном луче (вид со стороны передней панели)

Особенность автоматического управления режимом нагрева — применение адаптивного алгоритма, меняющего закон управления в зависимости от состояния процесса. Это позволяет учесть инерционность объекта и нагреть его за минимальное время с учетом ограничения на градиент температур.

Прикладное ПО системы (~4,5К байт) написано на языке Фокал. В режиме управления процессом от съема данных до выдачи управляющего воздействия проходит около 3 с.

Более совершенный вариант УСО дополнен схемой обмена цифровыми сигналами. Входной порт ПЭВМ делится на несколько направлений с помощью МБР (К5891Р12), запараллеленных по выходу и стробируемых сигналами ВК. Младшие десять разрядов порта вывода подаются на входные цепи нескольких ЦАП (К572ПА2А), а восемь младших разрядов делаются с помощью адресуемых по записи МБР. Для адресации регистров и задания режимов работы используются 10..14 разряды порта вывода. Дешифраторы адресов и сигналов управления выполнены на ИМС К155ИД3. 15-й разряд служит для задания направления обмена (прием-передача) и стробирует дешифраторы.

Этот вариант УСО позволяет создавать на основе «Электроники БК-0010» автоматизированные системы научных исследований в комплексе с серийными цифровыми приборами. Обладая графическими возможностями, она служит интеллектуальным цифровым осциллографом, измерителем частотных и вольт-амперных характеристик. В этих применениях для быстродействия в состав УСО включены буферное ОЗУ и быстродействующий АЦП.

На «Электронике БК-0010» создана система сбора данных о распределении энергии луча в электронно-лучевой установке (рис. 5). Кроме быстрых АЦП (К1107ПВ1) и ОЗУ (К541РУ2), аппаратно выполнен блок тестового сигнала. За время теста (1 мс) производится 1024 измерения отклика объекта. Параметры тест-сигнала и входные цепи устройства настраиваются программно. Данные после каждого цикла измерения переписываются в память машины, где проходит фильтрацию и вторичную обработку, заключающуюся в восстановлении трехмерной картины распределения энергии электронного луча по нескольким сечениям. Полученный результат выводится на экран дисплея в виде графика.

Для «Электроники БК-0010» разработан ряд систем подготовки программ. Для медленных объектов прикладное ПО целесообразно создавать на языке Фокал или Бейсик (включая драйверы УСО, диалог с оператором и графическую интерпретацию результатов).

Если задача легко разделяется на быструю и медленную составляющие или необходим экономичный формат хранения данных, то можно использовать программные системы ФОКОД или интерфейс внешних функций (см. сообщение А. Казанцева в этом же номере). Обе системы занимают небольшую часть ОЗУ и позволяют встраивать в программу на Фокале кодовые фрагменты.

Для быстродействующих объектов программирование стоит вести на языках Форт и Ассемблер, версии которых в адаптации для «Электроники БК-0010» будут приведены в следующих номерах журнала.

Адрес для справок: 107005, Москва, 2-я Бауманская ул., 5, МВТУ им. Н. Э. Баумана, Полянскому П. В.

ЛИТЕРАТУРА

1. Микропроцессоры. В 3-х кн. Кн. 1/Н.В. Воробьев, В. Л. Горбунов, А. В. Горячев и др.; Под ред. Л. Н. Преснухина.— М.: Высшая школа, 1986. С. 258—261.
2. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и мини-ЭВМ.— Л.: Энергоатомиздат, 1984, с. 227.
3. Вотинцев Н. Преобразователь напряжения с ШИ-стабилизацией//Радио.— 1985.— № 10.— С. 27.

РАДИОЭЛЕКТРОНИКА ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

В ВЫПУСКАХ ИЗДАТЕЛЬСТВА

«ЗНАНИЕ»

Читайте в наших выпусках:
о вычислительных системах
и программировании, о луч-
ших отечественных и зару-
бежных разработках, изо-
бретениях, открытиях, о ма-
шине с потоком данных, кон-
вейерных и матричных
ЭВМ новых поколений, оп-
тических и криогенных ма-
шинах, о создании супер-
ЭВМ, искусственного интел-
лекта, экспертных систем,
биокомпьютера



Читайте в наших выпусках:
о разработке отечественных
систем Телетекста и Видео-
текста, о производстве и
применении современной ин-
тегральной техники («сверх-
решетке», проблеме «тира-
нии соединений», рентгенов-
ской и ионной литографии),
о цифровых методах обра-
ботки информации, систе-
мах световодной и лазерной
передачи сообщений, спут-
никовой связи, о бытовой
электронике

Серия

Серия

• РАДИОЭЛЕКТРОНИКА • ВЫЧИСЛИТЕЛЬНАЯ
И СВЯЗЬ • ТЕХНИКА
И ЕЕ ПРИМЕНЕНИЕ •

70077

ИНДЕКС ИНДЕКС

70195

ЖУРНАЛЪ

ОБЩЕПОЛЕЗНЫХЪ СВѢДѢНІЙ.

7-е Сентября.

№ 37.

Третій годъ.

I. ВЕВЕДЖЪ И ЕГО ЧУДЕСНЫЯ ИЗОБРѢТЕНІЯ.

Бевбеджъ, Членъ Лондонскаго Королевскаго Общества, есть одинъ изъ людей, которыми Англія справедливо гордится, и который, подобно Ньютону и Гершелю, создасть своему гению прочный памятникъ.

Вотъ что говоритъ о немъ просвѣщенный путешественникъ, недавно посѣтившій Англію.

«Я навѣщалъ этого ученаго, еще молодаго и полного жизни. Въ головѣ его родилась смѣлая мысль, передавать, изображать колесами всѣ возможныя соотношенія чиселъ; онъ изобрѣлъ и оканчиваетъ теперь машину для вычисленія логарифмовъ, и въ то же самое время для гравированія ихъ на стали. Предложите моей машинѣ, говорилъ онъ мнѣ, уравненіе какой угодно степени, она въ минуту будетъ вамъ отвѣчать безошибочно. Я сдѣлалъ это, и машина, поставленная на нуль, начала дѣйствовать; квадратныя числа были возвыщаемы колокольчикомъ, другой означалъ корни, наконецъ дѣйствіе окончилось рѣшеніемъ столь же вѣрнымъ, какое могъ бы представить опытный математикъ, посвятивъ на то цѣлый день.»

«Эта единственная машина должна еще быть улучшена чрезвычайнымъ образомъ: я говорю единственная, потому что другой подобной быть не можетъ; Правительство приняло она себя издержки. Въ составъ ея войдетъ нѣсколько тысячъ колесъ, разнаго вида и устройства, отдѣленныхъ съ необыкновеннымъ совершенствомъ: она будетъ занимать мѣсто въ нѣсколько квадратныхъ саженей.»

«Понятно, что таблицы логарифмовъ, единожды выгравированныя, будутъ служить всемъ, а какъ въ составленіе ихъ не могутъ вовсе вкрасться ошибки, то довѣренность со стороны людей, ихъ употребляющихъ, будетъ общая.»

«Есть изобрѣтенія, говорилъ мнѣ Бевбеджъ,

«которыя могутъ быть выполнены только въ известное время. Мое, двадцать лѣтъ тому назадъ, было бы несовершеннымъ: одни только необыкновенные успѣхи Механики, и особенныя улучшенія въ токарномъ искусствѣ, дали мнѣ возможность осуществить эту мысль. — Я имѣю еще много замысловъ, боюсь ихъ представить на судъ общій; между тѣмъ я убѣжденъ, что исполненіе ихъ столь же вѣрно, столь же возможно, какъ исполненіе моей счетной машины. Такъ напримѣръ, я знаю, что изъ этой комнаты, занимаемой мною въ Лондонѣ, я могу разговаривать съ жителемъ Ливерпуля, и что мы будемъ слышать одинъ другаго на разстояніи 70 миль, какъ будто бы стояли бокъ о бокъ: нужно бы только было зарыть пустую трубку между двумя городами.»

Извѣстно, что Біотъ, Буваръ и Франкертъ, дѣлали подобныя опыты, на довольно большомъ пространствѣ, и говоря шопотомъ, слышали другъ друга, такъ хорошо, что, казалось, между ними вовсе не было пространства.

Бевбеджъ обнародовалъ свое средство, устроить чрезвычайно быструю почту, для писемъ, проволокъ, натянутыми на высокіе столбы. — Ему же принадлежитъ мысль, доставлять ихъ чрезъ широкій стволъ, заряженный цилиндромъ, наполненнымъ письмами, и гонимымъ въ немъ помощію дуговой машины; здѣсь скорость движенія могла бы быть болѣе миль въ двѣ минуты.

«Эти и многія еще изобрѣтенія останутся нашимъ внукамъ: для успѣховъ, вы видите, не можетъ быть ни границъ, ни мѣры.» Такъ говорилъ Бевбеджъ. Признаюсь, я удивлялся, что въ Англіи, какъ и во Франціи, умнѣйшіе люди жалуются на холодность, на равнодушіе, съ которыми принимаются самыя необыкновенныя изобрѣтенія; что и тамъ, какъ у насъ, болѣе нежели скупы на средства, необходимыя для приведенія ихъ въ исполненіе!

Рд.

Данную технологию построения БЗ кросс-системы можно развить для своего проекта, создав библиотеку наиболее часто применяемых программ и оформив их как объектные модули, сопряженные между собой макроопределениями. Это составит ваш проблемный язык, основывающийся на «базе хранения проектов и апробированных модулей» (см. рис. 1).

Характеристика кросс-системы РАСА-84, построенной по принципу БЗ для КР580ИК80.

Наполнение БЗ РАСА-84 можно охарактеризовать так (рис. 3):

1. Кросс-система проектирования ПО, предназначенная для программирования МПС на базе МПК КР580, реализуется в среде ОС ДВК средствами МАКРО-11.

Ядро БЗ РАСА-84 можно переписать в любую другую ОС, в которой функционирует МАКРО-11.

2. Программирование разделено на три уровня:

конструктивное — описание решаемой МПС задачи в целом в виде взаимодействия процессов, потоков данных в процессах и распределения ресурсов;

системное — описание алгоритмов связи внутренних потоков данных и событий с внешней средой;

проблемное — описание процессов обработки данных и событий, связанных с внешней средой, задан-

ных на конструктивном уровне и согласованных с протоколами обмена данными, задаваемыми на системном уровне.

3. Вспомогательные средства генерации ПО включают в себя резидентные отладчики, загрузчики, модели внешней среды МПС и средства управления ими.

4. Объектный образ задачи отлажен на модели в МПС.

5. Модули ПО, генерируемого системой, можно использовать совместно с модулями, полученными в других системах проектирования программ для КР580 при условии введения в систему соглашений о программных интерфейсах и форматах данных.

6. Систему можно использовать с аппаратурой отладки типа встроенного эмулятора, которые дают точные приближения к реальному процессу для однопроцессорных систем.

Проблемы и направления развития.

Итак, в РАСА-84 заложены сведения о конструировании и программировании МПС, с которыми можно манипулировать, создавая свои модели. Однако чем сведения обширнее, тем труднее становится работать с ними в автоматизированном режиме, так как то, что выражено лексической единицей, надстроенной над стандартной мнемоникой команд МП, обозначающей оператор, одно-

му понятно (так как он сам вносил в БЗ эти знания), а для другого это пустое слово, не несущее нагрузки без описания его значения.

Вследствие этого при данном подходе к построению кросс-систем возникает проблема развития управляющей системы БЗ (которую удобно построить на основе реляционной базы данных (РБД) в рамках РТК-МИКРО).

Вопросы ориентации в БЗ и эффективного манипулирования знаниями — наиболее сложные при создании БЗ, так как требуют для своего решения привлечения лингвистики и инженерной психологии. В частности, с этими проблемами связано слабое распространение технологии знаний в области конструирования ПО. Для больших универсальных ЭВМ такие системы созданы и совершенствуются [4, 5]. В среде ДВК уже сейчас можно построить приемлемые средства для БЗ о микропроцессорном комплексе, используя РТК-микро [6], в частности, РБД для создания тезауруса МПК и механизмов МЕНЮ для реализации последовательности технологических процедур. Тезаурус на базе РБД будет представлять собой таблицу, в которой размещаются следующие параметры:

<[Имя]МПК> — это термин, который используется в БЗ,

<[Имя]осн[овного]тер[мина]> — это термин, частью которого является указанный термин,

<Акт[ивность]> — название части БЗ, в которой термин используется активно,

<Исх[одный]тек[ст]> — место хранения исходного текста,

<Инстр[укция]> — место хранения неформального описания,

<Гот[овность]> — код готовности указывает, какие стадии обработки в БЗ термин прошел,

<Ком[ментарий]> — краткая аннотация термина.

Все эти параметры образуют следующую реляционную таблицу:

<МПК><осн тер><Акт><Исх тек>
<Инстр><Гот><Ком>

Наполнение таким образом заданной РБД становится словарем-справочником по проблеме конструирования ПО для МПК в созданной БЗ при условии того, что любое новое знание, вносимое в БЗ, будет отображаться в виде терминов в РБД.

Отметим также, что на основе построения БЗ РАСА-84 мы сформулировали общие требования к средствам организации БЗ для моделирования широкого круга МПК. Эти средства можно получить созданием следующих универсальных программ.

1. Текстовый процессор для формирования и работы с БЗ, обладающий возможностями задавать:

а) лексические единицы в виде макроопределений (как, например, МАКРО-11);

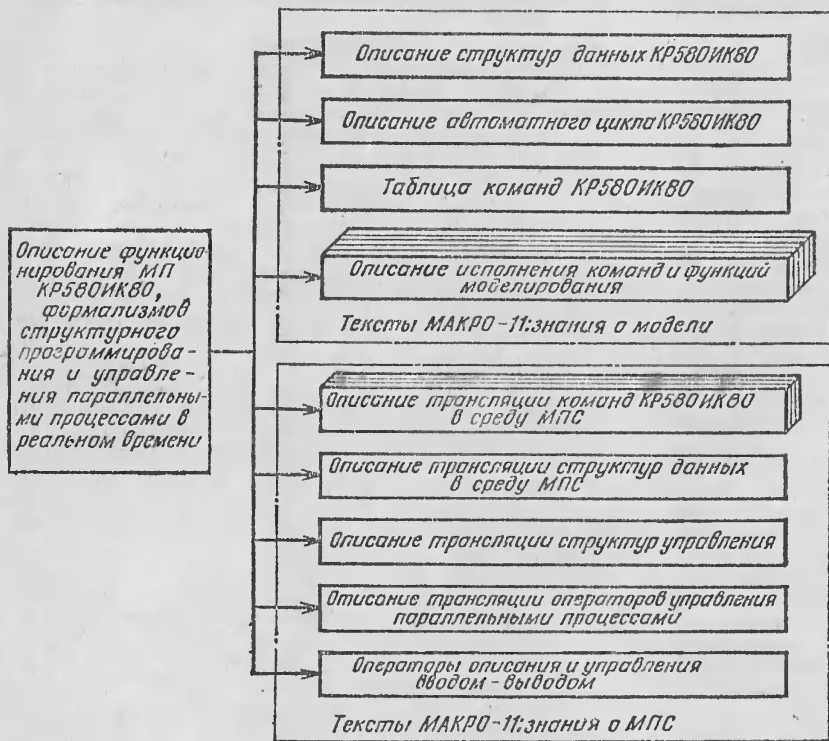


Рис. 3. Схема преобразования неформальных описаний, использовавшихся при построении РАСА-84, в знания, закодированные в виде текстов МАКРО-11

б) в макроопределениях лексических единиц одновременно несколько различных типов расширений, как то:

двоичное расширение с привязкой к ранее определенным счетчикам двоичных слов (разрядность слов также декларируется отдельно),

различные типы текстов (в частности, рекурсивные), параметры лексем для словарной службы проекта (ведение тезауруса проекта);

в) грамматические правила связывания лексических единиц в тексты (это требование может быть ослаблено использованием стандартной грамматики написания текста над и под дугами в рамках ГОСТ 19.05—85 на Р-технологии);

г) правила связывания для объектных (двоичных) текстов;

д) текстовый процессор также может вести словарь лексем и управлять работой с ним.

2. Система управления инкрементным вводом и редактирования тек-

стов, словаря, объектных (двоичных) структур технологического цикла.

3. Система адаптации, включающая в себя текстовые компаратор, трансформатор, управляемый интерактивно, для форматных преобразований текстов в заданной грамматике входного и выходного форматов, отладчик.

Все указанные программы можно реализовать на основе уже существующих спецификаций на средства, описанные в статье. На наш взгляд, построение подобного комплекса технологических средств на достаточно мощной ЭВМ позволит не только ускорить процесс освоения МПК, но и создать новые концепции в области систем команд и архитектуры МПК и всесторонне исследовать их до создания заказных БИС.

Телефон для справок — 124-74-04 (Москва).

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иваников А. Д. Проекти-

рование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.

2. Лукьянов Д. А. Как написать кросс-транслятор с языка ассемблер // Микропроцессорные средства и системы.—1985.— № 4.— С. 35—41.
3. Янг. Алгоритмические языки реального времени.— М.: Мир, 1985.
4. Бабаев И. О., Новиков Ф. А., Петрушина Т. И. Декарт — входной язык системы СПОРА.— М.: Прикладная информатика, 1981, вып. 1.
5. Кахро М. И., Калья А. П., Тыгу Э. Х. Инструментальная система программирования на ЕС ЭВМ (ПРИЗ)— М.: Финансы и статистика, 1981.
6. Вельбицкий И. В., Кованев А. Л. Графический стиль программирования для персональной ЭВМ // Микропроцессорные средства и системы.—1985.— № 4.— С. 46.

Статья поступила 15 декабря 1986 г.

УДК 681.3.06

С. И. Берестихевский, Т. В. Колосова, О. Н. Мартыненко

ПАКЕТ ПРОГРАММ ПО ПРИКЛАДНОЙ СТАТИСТИКЕ ДЛЯ ПЕРСОНАЛЬНЫХ ЭВМ

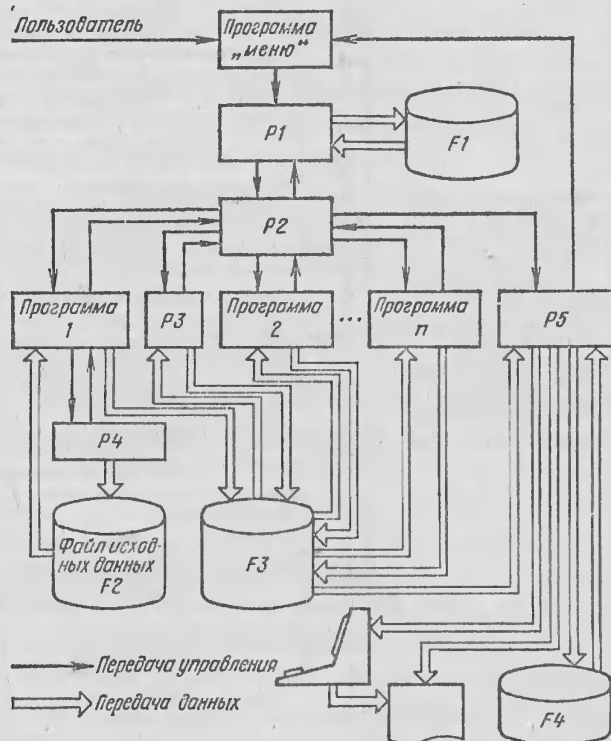
Предлагаемый для использования пакет программ по прикладной статистике (ППС) предоставляет удобное и единообразное обращение к любому методу обработки данных с помощью иерархически построенного «меню» (и последовательное использование нескольких методов обработки данных); организацию исходных данных в виде файлов на магнитных дисках со средствами их ведения; простую систему файлов, хранящих, в частности, результаты обработки, и позволяющую выходу одного метода служить в качестве входа последующих методов обработки; средства вывода результатов обработки в алфавитно-цифровом и графическом виде на дисплей и печатающее устройство, запрограммированное в режим плоттера, не используя специальных языков запроса пользователя.

При работе с пакетом ППС от пользователя не требуется знаний какого-либо языка программирования.

В пакете ППС организованы четыре тематических раздела: предварительный статистический анализ данных; статистический анализ временных рядов; оценивание параметров фиксированных структур математических моделей; выборочный контроль качества, что важно для служб Госприемки. Пакет состоит из программ, реализующих 32 процедуры прикладной статистики, сервисных программ, средств взаимодействия с пользователем и защиты от «недоброжелательного» доступа. При использовании пакета ППС его пользователь найдет ответы на вопросы о том, как готовить данные для машинной обработки, как проводить разведочный анализ данных с целью формирования гипотез относительно их вероятностной структуры, какие виды анализа можно проводить на ЭВМ в результате применения

того или иного метода, каковы идея каждого метода анализа и границы его применимости.

Взаимодействие в системе «Пользователь — ЭВМ — программное средство» обеспечивается информационной структурой пакета ППС, организованной в виде параллельного программного интерфейса. Функционирует информационная структура (см. рисунок) таким образом: программа «меню» предоставляет пользователю воз-



Информационная структура пакета прикладной статистики

возможность выбора прикладных программ и указания последовательности их выполнения. Программа P1 записывает последовательность выбранных прикладных программ в файл F1. Программа P2 управляет выполнением последовательности прикладных программ (указанных в файле F1), а также программой P3, передающей данные между прикладными программами, и программой P5, обеспечивающей вывод результатов. Прикладная программа, выбранная пользователем, запрашивает у него имя входного набора данных, создание и ведение которого обеспечивается программой P4. Программа P3 под управлением программы P2 создает и ведет файл F3 передачи данных между прикладными программами. Программа P5 под управлением программы P2 создаст файл результатов F4 и управляет выводом результатов на дисплей и печатающее устройство.

Пакет ППС эксплуатируется на персональных ЭВМ стандарта MSX, имеющих ОЗУ не менее 64К байт и дисковое ЗУ не менее 500К байт. Пакет ППС работает под управлением интерпретатора MSX-BASIC.

Развитие и совершенствование работ в области программного обеспечения ЭВМ по прикладной статистике

УДК 681.3.06

В. Э. Хаер

ПАКЕТ ПРОГРАММ ДЛЯ ХРАНЕНИЯ ПЛАНОВО-ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ НА МИКРОЭВМ «ИСКРА 226»

Программное обеспечение для обработки планово-экономической информации на микроЭВМ «Искра 226» разрабатывается, как правило, на основе систем табличной обработки данных (СТОД), таких как «Нива» [1], «Дисота» [2], «Прокт» [3]. Поддерживаемые СТОД структуры данных — двумерные матрицы — удобны для выполнения на ПЭВМ процедур ввода и коррекции информации, расчетов, печати документов, но мало подходят для организации больших баз данных (БД), обладающих свойствами минимальной избыточности, целостности и независимости данных.

Пакет программ для хранения планово-экономической информации «Ирис» расширяет возможности СТОД, позволяя вести БД, состоящую из «атомарных» единиц экономической информации — показателей, и формировать из их значений таблицы, необходимые для обработки.

Структура пакета прикладных программ (ППП) представлена на рис. 1.

База данных состоит из информационных объектов трех типов — словаря, связей и показателей, достаточных для хранения информации о любой экономической системе.

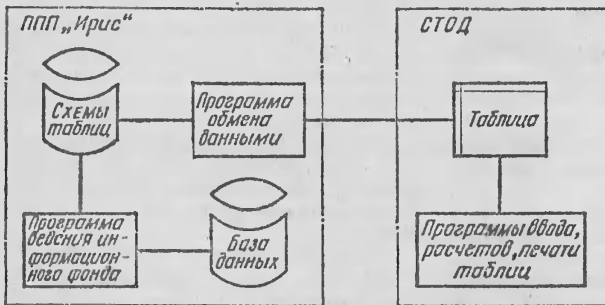


Рис. 1. Структура ППП «Ирис»

чрезвычайно важно, так как аппарат прикладной статистики — это математическая основа построения элементов автоматизированных информационных систем, разработки элементов САПР и роботизации промышленности, анализа и оптимального управления технологическим процессом, прогноза социально-экономических характеристик, медицинской диагностики и др.

Пакет ППС открыт для расширения за счет введения новых прикладных программ. В настоящее время в пакет ППС включаются методы классификации многомерных наблюдений, основанные на кластер-анализе и методе динамических сгущений.

Пакет ППС апробирован при моделировании технологического цикла получения пищевого белка из сока зеленых растений и при анализе данных из области социально-гигиенических проблем для прогнозирования степени риска возникновения заболеваний.

Адрес для справок: Запорожье, ул. Жуковского, 66, Запорожский гос. ун-т, Кафедра информатики, тел. 64-54-49.

Статья поступила 25 ноября 1986 г.

В словаре перечислены все экономические объекты, о которых в БД может накапливаться информация. Для каждого объекта заданы имя, идентифицирующее объект для пользователя, и код, используемый для ссылок на объект системой.

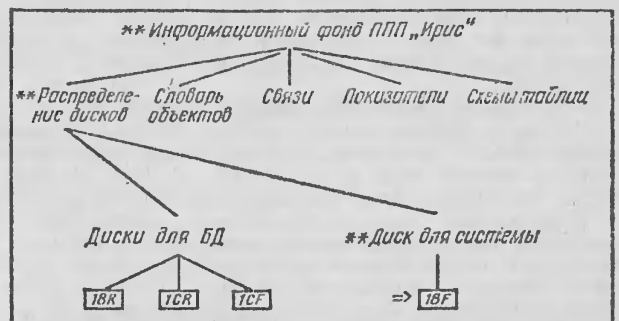
Связь представляет собой многозначную функцию, сопоставляющую объектам множества объектов. Связи используются для фиксации парадигматических (условно-постоянных) отношений между объектами, выражаемых с помощью перечней, номенклатур, классификаторов.

Показатель — это функция нескольких аргументов, сопоставляющая набором объектов числовые значения характеристик.

Схема таблицы задает способ формирования таблицы из связей и показателей БД.

Программа ведения информационного фонда обеспечивает: создание, коррекцию и печать информационных объектов БД и схем таблиц; восстановление разрушенных в результате сбоя файлов информационного фонда; изменение физического представления БД для оптимизации ее функционирования.

Все виды работ выполняются в диалоговом режиме. Диалог с программой основан на принципе непосредственного редактирования. Оператор представляет информационный фонд в виде дерева, с внутренними



Обозначения: □ — значения данных; => — курсор; ** — след курсора (путь из текущей вершины в корневую).

Примечание. Поддерево «распределение дисков» выводится на экран только при нахождении курсора в его пределах.

Рис. 2. Фрагмент дерева диалога

вершинами которого связаны имена, а с листовыми — значения данных. На экране дисплея (рис. 2) изображается видимый из текущей вершины фрагмент дерева диалога. С помощью команд перемещения по дереву — вверх, вниз, вправо, влево — оператор подводит курсор, указывающий на текущую вершину, к интересующим его данным и при необходимости изменяет их значения.

Программа обмена данными выполняется под управлением распространенной СТОД «Дисота» и обеспечивает чтение и запись значений показателей из БД в таблицу по заданной схеме таблицы. В пакет могут быть включены программы обмена данными с другими СТОД.

Технические характеристики пакета: язык программирования — Бейсик 2; объем программ — 150К байт; время формирования таблицы из 500 элементов — 8...12 мин; объем внешней памяти, расходуемой на хранение одного ненулевого значения показателя, — 16 байт; объем документации ЕСПД — около 100 листов.

Пакет программ «Ирис» разработан в Отраслевой

лаборатории по АСУ Московского лесотехнического института.

Телефон для справок: 582-47-50, Москва.

ЛИТЕРАТУРА

1. Бессонов В. Ф., Рыбаков В. И. Универсальная диалоговая система проведения плановых расчетов «Нива»/Сб. статей «Вопросы создания АСПР». — М.: ГВЦ Госплана СССР, 1983. — Вып. 56. — С. 111—120.
2. Собельман В. И., Шахвердов В. А. Диалоговая система обработки таблиц (Дисота) для мини-ЭВМ «Искра 226»/Сб. статей «Прикладная информатика». — М.: Финансы и статистика, 1987, вып. 1.
3. Казаков А. М. Описание возможностей системы «Проект» и методики ее использования/Сб. статей «Автоматизация расчетов к проектам планов экономического и социального развития». — М.: ЦЭНИИ и НИИ АСУ при Госплане РСФСР, 1983.

Статья поступила 5 февраля 1987 г.

УДК 681.325

А. Д. Горожкин, Л. А. Вознюк, Д. А. Чвыров

ОПЕРАЦИОННАЯ СИСТЕМА СР/М НА БАЗЕ КНМЛ

МикроЭВМ «Электроника К1-10», «СО-01» и другие содержащие в качестве центрального процессора микропроцессор КР580ВМ80А, имеют в базовом варианте внешнюю память с перфоленточным носителем, однако допускают относительно простое расширение конфигурации за счет подключения, например, кассетных накопителей на магнитной ленте (КНМЛ) через интерфейсные БИС КР580ВВ55.

Применение КНМЛ СМ5211 в качестве внешней памяти в указанных микроЭВМ, кроме таких очевидных достоинств, как снижение времени доступа к информации и повышение надежности носителя, позволило реализовать операционную систему СР/М 2.2. (Так же можно выполнить и другие версии СР/М: ОС 1800 и микроДОС.)

Особенностью реализации является включение в базовую систему ввода-вывода (BIOS) программы, эмулирующей магнитный диск на магнитной ленте. Программа-эмулятор выполняет блокирование секторов перед записью на ленту, позиционирование ленты при чтении или записи и деблокирование секторов после чтения информации. После «холодного старта» системы в специальный буфер ОЗУ считывается оглавление кассеты, и все операции с оглавлением осуществляются в ОЗУ. Измененное оглавление записывается на кассету при «теплом старте» системы. Программа-эмулятор полностью сохраняет все подпрограммы доступа к диску. Это позволяет использовать прак-

тически любое программное обеспечение, работающее в среде СР/М. Ускорение поиска информации достигается предварительной разметкой ленты. Емкость одной стороны размеченной кассеты составляет 256К байт. Возможно также применение бытового магнитофона типа «Маяк-231» или подобного с небольшими аппаратными доработками.

Опыт эксплуатации микроЭВМ «Электроника К1-10» и «СО-01» совместно с КНМЛ СМ5211 в учебных лабораториях кафедры ВТ КПИ показали высокую надежность такой конфигурации, хотя производительность кассетных накопителей существенно ниже, чем у накопителей на гибких магнитных дисках (НГМД). Замедление доступа к информации по сравнению с НГМД мало сказывается при работе таких системных программ, как ED, ASM, SID, POWER, MBASIC, и является существенным при использовании трансляторов с языков высокого уровня, имеющих оверлейную структуру.

Применение КНМЛ в качестве базового носителя СР/М обеспечивает не только копирование содержимого «электронного диска» на сменном носителе, но и позволяет сохранять работоспособность системы даже при отказе этого диска.

За справками обращаться по адресу: 252056, Киев, просп. Победы, 37, Киевский политехнический институт, каф. ВТ, корп. 18, к. 509, тел. 441-95-66.

Статья поступила 1 декабря 1986 г.

ЭТО МОГ ЧИТАТЬ ПУШКИН

Как вы думаете, когда в России впервые была опубликована информация о программируемой вычислительной машине? Оказывается в 1835 году в «Журнале общепольных сведений». Это мог читать Пушкин! Любопытно, правда? В моем распоряжении имеется оригинал. Ниже вы увидите текст оригинала (см. 4 стр. вкладки).

Возникает несколько вопросов: кто такой «Рд.» — автор публикации и кто такой «Путешественник»? Сообщил ли «Путешественник» сведения лично «Рд.» или статья является переводом из иностранного журнала? Если верно последнее, то из какого? Может быть существуют еще более ранние публикации?

Не пора ли организовать в СССР музей электроники и вычислительной техники? Скоро упадет на свалку последняя БЭСМ-6 — машина с рекордной живучестью, составившая эпоху в советской вычислительной технике. Первые советские ЭВМ уже утрачены. Скоро будут окончательно утрачены первые советские микросхемы. А где найти первый советский транзистор П-1? Если музей не будет организован в ближайшие год-два, то окажется поздно.

Я готов безвозмездно передать в распоряжение Музея электроники и вычислительной техники указанный выше номер «Журнала общепольных сведений».

Этот журнал — еженедельник, каждый номер состоит из 5—6 листов обычного журнального формата. По стилю — что-то вроде «Науки и жизни».

Адрес для справок: 141980, Дубна, Моск. обл., ул. Строителей 10-49, Телефон: 46-583, Федюнькин Евгений Дмитриевич

ИНЖЕНЕР ОФОРМЛЯЕТ ЗАЯВКУ НА ИЗОБРЕТЕНИЕ

XXVII съезд партии поставил перед коллективом Государственного комитета СССР по делам изобретений и открытий общую задачу: резко улучшить качество работ по заявкам на изобретения и в том числе ликвидировать случаи неправомерной задержки с принятием окончательных решений по заявкам на изобретения — ускорить принятие окончательных решений. Дальнейшее совершенствование государственной научно-технической экспертизы заявок на изобретения в условиях жестких ограничений численности работающих ИТР немислимо без коренного изменения самой «технологии» экспертизы: необходим качественно новый уровень технологии патентной экспертизы — на базе вычислительной техники, средств связи и ортехники.

Всесоюзный научно-исследовательский институт государственной патентной экспертизы, Государственный научно-исследовательский институт радио и творческие группы специалистов других организаций провели инициативную работу по исследованию возможности разработки сети безбумажной технологии создания, передачи, оперативного хранения и обработки документов по заявкам на изобретения. Публикуемые статьи дают представление о некоторых, на наш взгляд, принци-

пиально важных технических решениях затронутой проблемы.

Член коллегии Госкомизобретений СССР
Начальник отдела экспертизы изобретений
В. М. КАЗАНКОВ

От редакции. Кто из нас не испытывал чувства авиапассажира, у которого перелет из одного города в другой занимает гораздо меньше времени, чем процесс получения багажа, всевозможные регистрации, переезд из аэропорта в город и т. п. В аналогичном положении находится сегодня изобретатель, у которого время зарождения идеи изобретения и даже работа по созданию макета и его испытание занимает, как правило, меньше времени, чем процедура поиска аналогов и прототипа, написание текста заявки, подготовка и печатание необходимой сопроводительной документации. Судя по редакционной почте, для многих наших читателей представляют практический интерес результаты разработки такого инструмента, который переложил бы на плечи мини-ЭВМ эту рутинную часть многотрудного изобретательского творчества.

УДК 681.322.1—181.4

В. К. Сарьян, Г. Г. Смслич

ВЗАИМОДЕЙСТВИЕ БОЛЬШИХ И МАЛЫХ БАЗ ДАННЫХ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ ОФОРМЛЕНИЯ ЗАЯВОК, ОРИЕНТИРОВАННОЙ НА ИНДИВИДУАЛЬНЫХ ПОЛЬЗОВАТЕЛЕЙ

Временной цикл оформления заявки на предполагаемые изобретения складывается из следующих основных этапов: зарождение у изобретателя идеи изобретения, классификация изобретения и определение класса по МКИ (Международной классификации изобретений) аналогов; изучение аналогов и выбор прототипа (это позволяет ясно сформулировать сущность своего технического предложения). Наиболее сложен для автора предполагаемого изобретения перевод идеи технического предложения в формулу. После этого автором оформляются полный текст заявки и первичные сопроводительные документы, как правило, в сотрудничестве с патентоведам предприятия.

Полный комплект документов по заявке передается во Всесоюзный научно-исследовательский институт государственной патентной экспертизы (ВНИИГПЭ). На всех этапах цикла оформления заявки изобретатель должен строго придерживаться нормативных документов, которые часто меняются. Кроме того, на отдельных этапах работы с патентоведами предприятия или ВНИИГПЭ он должен пересмыслить по вновь получаемой информации ранее принятые решения, в том числе о способе и объеме правовой охраны своей идеи.

Анализируя этот цикл, можно заметить, что наличие Положения, нормативных актов и указаний, строго регламентирующих форму и содержание первичных документов по заявкам и решений экспертизы, позволяет формализовать значительную часть процесса подготовки материалов по заявке. Можно также решить задачу автоматического накопления опыта экспертизы заявок по конкретным предметным областям творческой деятельности инженера, вполне созревшим для автоматизации с позиции «феномена деловой прозы» [1] и позиции автоформализации профессиональных знаний [2].

В работе [3] описаны принципы первой реализации автоматизированной системы оформления первичных документов по заявкам на предполагаемые изобре-

тения. Она выполнена в 1984—1986 годах в Государственном научно-исследовательском институте радио на базе отечественных микроЭВМ «Искра 226», «Электроника 60», «Электроника НЦ 801 ДМ, НЦ 801 ДА», ДВК-2 и др.

Разделение текста всех первичных документов на информационно-самостоятельные части и определение взаимосвязи между этими частями позволяет выделить информационно-независимые части (в дальнейшем — исходные данные) и определить процедуру автоматизированного совмещения этих исходных данных (непосредственно или с соответствующей синтаксической трансформацией) с формой необходимой реализующей структуры. Определение взаимосвязи позволяет также составить программу для автоматизированной формальной проверки по введенному в ЭВМ полному тексту заявки на правильность выполнения автором рекомендаций многоуровневой подсказки и смысловой проверки на основе «модели эксперта».

Исходные данные в системе

- 1) название объекта изобретения (способ, устройство и т. п.);
- 2) фамилия, имя и отчество автора(ов);
- 3) название заявки;
- 4) класс последней редакции МКИ, относящийся к данному названию изобретения;
- 5) ограничительная часть, цель и отличительная часть с обязательным выделением в ограничительной и отличительной частях отдельных признаков;
- 6) номер УДК;
- 7) описание и недостатки аналогов;
- 8) библиографические данные прототипа и аналогов;
- 9) данные о базовом объекте.

Кроме того, автор вводит с клавиатуры дисплея информационно-самостоятельные части: описание работы в динамике и технико-экономические показатели заяв-

ляемого технического решения по сравнению с базовым объектом.

По этим данным система формирует текст заявки и первичные документы по ней.

Анализ оформления заявок на изобретения с помощью этой системы позволил бы еще больше сократить сроки оформления за счет автоматизации поиска аналогов и прототипа, а также подбора базового объекта. Можно автоматизировать и занесение статистических данных по заявкам в журнал регистрации или в базу данных.

Для решения этого вопроса индивидуальному пользователю крайне неудобно применять имеющиеся базы данных (по патентной и непатентной информации) в ВНИИПИ, ВИНТИ и др., так как он удален от этих баз, а доступ к ним ограничен по времени. Информация приносит наибольшую пользу, если она поступает в нужный момент, и, таким образом, мы приходим к выводу о необходимости создания на каждом предприятии своей базы данных. Своя БД на каждом предприятии необходима еще и потому, что содержимое базы должно отражать тематику предприятия, а избыточная информация замедляет реакцию системы.

Кроме информации, поступающей из информационных Всесоюзных центров, для ведения патентной работы на предприятии необходимы конъюнктурная информация и статистические данные. Так как эти три вида информации несходны, то на предприятии должны быть три больших базы данных (функционируют на ЭВМ большой или средней мощности).

БД 1-го рода — это база данных, содержащая фрагменты патентной и непатентной информации (поступают централизованным порядком из Всесоюзных информационных центров на магнитных носителях). Достоинством для работы единица патентной и непатентной информации — реферат из реферативных сборников ВНИИПИ, ВНИИТИ, ВНИИТЦ и других организаций [4]. Небольшая информационная емкость рефератов позволяет создавать БД 1-го рода на ЭВМ средней или большой мощности, имеющихся в настоящее время у большинства предприятий. При этом нет большого ущерба решению других задач на ЭВМ. Получаемую информацию можно преобразовать в удобный для использования на предприятии вид. Из этих организаций информация в БД 1-го рода поступает не непрерывно, а по мере накопления, фрагментами. Подчеркнем, что информация в этой БД не должна изменяться и замещаться индивидуальными пользователями.

В БД 2-го рода собирается и хранится конъюнктурная информация. Эта информация может поступать как извне, так и в результате патентно-лицензионной деятельности индивидуальных пользователей и работников патентной службы на данном предприятии. Необходимые атрибуты конъюнктурной информации — источник поступившей информации и фамилия, имя, отчество пользователя, внесшего эту информацию.

Заметим, что к конъюнктурной информации относятся данные об объектах техники, их технико-экономические показатели (в том числе и объектов, основанных на изобретениях), данные о рынках сбыта объектов техники и т. п. Эти данные далеко не всегда можно получить со стороны. Кроме того, полученные со стороны данные могут быть преднамеренно искажены, поэтому они накапливаются в БД 2-го рода в результате кропотливой работы по отбору и фильтрации научно-технической информации из различных источников, из личных контактов со специалистами и т. д.

В БД 3-го рода автоматически собираются и хранятся статистические данные по заявкам индивидуальных пользователей на индивидуальные автоматизированные системы: фамилия, имя и отчество автора заявки, подразделение, в котором работает автор (авторы), номер НИР или ОКР,

в ходе которых создана заявка, название изобретения, МКИ, УДК, экономический эффект и т. п.

Между БД 1-го и 2-го рода могут быть общие реферативные, по которым возможен поиск и получение обобщенной информации: название фирмы или учреждения заявителей данных патентов и авторских свидетельств и др.

Такие БД позволяют изобретателю сократить время написания заявки. Недостатки подхода — функционирование этих БД на машинах большой и средней мощности заставляет изобретателя заранее планировать время общения с ЭВМ. Это не способствует максимальной творческой отдаче, так как изобретателю нужен инструмент «сиюминутно», когда мысль «созрела». Кроме того, ему не нужна вся информация, хранящаяся в больших БД (она тормозит его работу как избыточная). Изобретателю также необходимо накапливать, перерабатывать, обобщать и хранить свой опыт, нужную ему информацию, не раскрывая своих «секретов» до получения охранных документов. При работе над заявкой автор может неоднократно обращаться к фрагменту информации, но при работе на большой ЭВМ ему надо планировать время общения с машиной, что «выбивает из колеи».

Уровень развития вычислительной техники позволяет говорить о том, что изобретатели в каждом небольшом подразделении должны работать с персональной ЭВМ. Итак, для наиболее эффективной творческой отдачи изобретателю необходимо дать персональный инструмент со своими малыми базами данных. По аналогии с большими БД их можно классифицировать следующим образом.

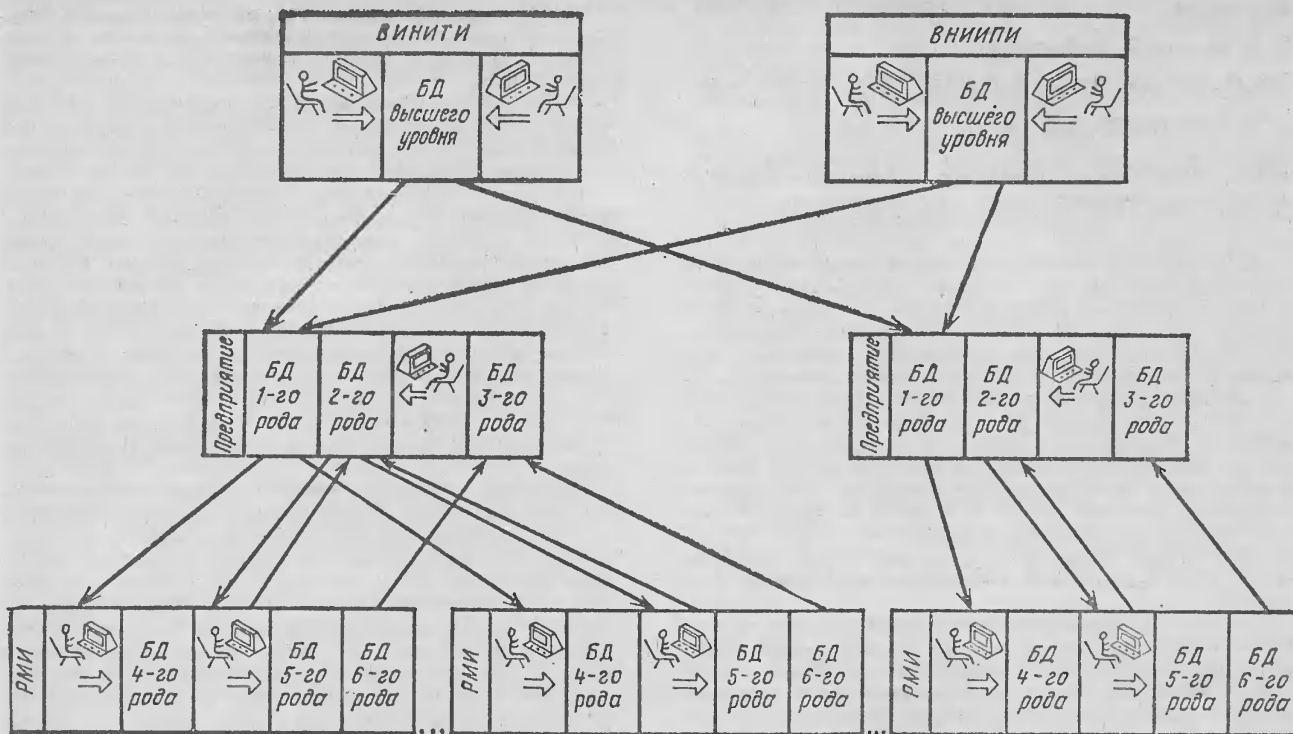
БД 4-го рода — это узкий по конкретной тематике, но полный по содержанию фрагмент информации, ориентированный на конкретного разработчика (группу разработчиков, работающих по одной тематике). Информационные единицы этого фрагмента в ходе работы системы могут замещаться переработанным содержанием. Разработчик получает эту информацию из БД 1-го рода, перерабатывает ее к удобному для него виду, переосмысливает ее. С течением времени новые фрагменты из БД 1-го рода вытесняют устаревшую информацию из БД 4-го рода. Необходимо отметить, что никакая информация из БД 4-го рода в БД 1-го рода не возвращается, БД 4-го рода будет использоваться индивидуальным пользователем при оформлении заявок для автоматизированного выбора и формирования таких исходных данных, как «ограничительная часть», «описание и недостатки аналогов», «библиографические данные аналогов и прототипа».

БД 5-го рода — в начальный момент это фрагмент БД 2-го рода. Особенность ее — ориентация на конкретного разработчика, она содержит сведения, добытые им лично в процессе работы или аналитической переработки информации, т. е. отражает его активность и профессиональную компетентность. Фактические данные этой БД могут с согласия индивидуального пользователя передаваться с указанием, от кого они поступают (источника информации) в БД 2-го рода. В БД 2-го рода они могут поступать в режиме накопления и в режиме замещения.

Содержимое БД 5-го рода используется для автоматизированного выбора и формирования исходных данных о базовом объекте (название объекта, завод-изготовитель, основные технические параметры объекта и т. п.).

БД 6-го рода — это база данных, которая накапливает статистические данные по оформленным на персональной системе заявкам, необходимые для функционирования системы делопроизводства по заявкам, контроля на предприятии за изобретательской деятельностью. Информация из всех БД 6-го рода передается и накапливается в БД 3-го рода.

При таком построении малых баз данных одна из



РМИ — рабочее место изобретателя, организованное на базе мини-ЭВМ

функций автоматизированных систем, ориентированных на индивидуальных пользователей,—использование фрагментов этих БД при написании и оформлении заявки, переработке этих фрагментов и подготовке данных к передаче их в БД более высокого уровня.

Подчеркнем, что нами выделено три уровня баз данных (см. рисунок), Высший—это БД Всесоюзных информационных центров, заполняемые сотрудниками этих учреждений по документам, получившим юридическую силу.

Средний уровень (базы данных предприятия) — это БД 1-го, 2-го и 3-го родов. БД 1-го рода пополняется только из баз данных высшего уровня, а БД 2-го рода,—как правило, из БД 5-го рода, а также сотрудниками предприятия, имеющими санкционированный доступ. Заметим, что передача информации из БД 5-го рода в БД 2-го рода может быть только санкционированной; при этом наряду с доверительным коэффициентом необходим и «поощрительный». Пользователь, внесший большее число достоверных данных, заслуживает материального и морального поощрения. БД 3-го рода пополняется из всех БД 6-го рода. База данных среднего уровня — аккумулятор всех низших соответствующих баз.

Низший уровень (индивидуальные базы данных) — БД 4-го, 5-го и 6-го родов. БД 4-го рода формируется самим пользователем из БД 1-го рода в удобном для него виде, на привычном для него языке. Просматривая за дисплеем тематику БД 1-го рода, пользователь может периодически выбирать для себя в свою базу низшего уровня новую поступившую информацию. Между БД 2-го и 5-го родов ведется двусторонний обмен в режиме взаимного обогащения. БД 6-го рода пополняется исключительно пользователем, и информация из нее в одностороннем порядке передается в БД 3-го рода.

Нами затронуты только некоторые вопросы организации данных и обмена ими в изобретательской, патентной и лицензионной деятельности. Пока не рассмотрены согласования СУБД различных предприятий, а также СУБД для малых и больших ЭВМ. Не менее важна структура используемых данных. Она существенно влияет на эффективность функционирования автоматизированной системы оформления заявок на предполагаемые изобретения.

Телефон для справок — 261-11-09 (Москва).

ЛИТЕРАТУРА

1. Ершов А. П. К методологии построения диалоговых систем: феномен деловой прозы. Препринт № 156. — Новосибирск, ВЦ СО АН СССР, 1979.
2. Громов Г. Р. Автоматизация профессиональных знаний // Микропроцессорные средства и системы. — 1986. — № 3. — С. 80—91.
3. Сарьян В. К., Александров В. А., Смолнич Г. Г., Ефимова Л. Г. Использование реализующих структур для оформления первичных документов по заявкам и формализованная проверка описаний к изобретениям // Материалы III науч.-техн. конф. «Проблемные вопросы совершенствования системы выявления и защиты советских изобретений в процессе государственной научно-технической экспертизы». — М.: Госкомизобретений, 1986, с. 96—101.
4. Сарьян В. К. Опыт комплектования, хранения и использования патентного фонда в НИИ и тенденции его развития // Материалы Всесоюз. науч.-техн. конф. «Патентная информация на службе технического прогресса». — М.: ВНИИПИ, 1986, с. 236—240.

Статья поступила 10 января 1987 г.

В. К. Сарьян, В. В. Корытов

УВЕЛИЧЕНИЕ ВРЕМЕНИ ЖИЗНИ ДИАЛОГОВОЙ СИСТЕМЫ ПО ОФОРМЛЕНИЮ ПЕРВИЧНЫХ ДОКУМЕНТОВ ПО ЗАЯВКАМ

Эффективность диалоговых систем часто зависит от степени формализации процессов переработки информации в рамках решаемой задачи. В связи с этим можно ожидать высокую эффективность диалоговых систем, используемых для оформления первичных документов по заявкам на авторские свидетельства [1, 2].

Прикладные программы имеют короткое время жизни: программа обычно устаревает в течение года [3]. Опыт эксплуатации системы [1] показал, что время жизни диалоговых систем коротко из-за негибкости системного и программного обеспечения при взаимодействии с внешней средой (изменение во внешней среде требует изменения программного обеспечения), с конкретным пользователем и конкретной технологией (нет возможности оперативно перестроить систему).

Внешние среды для диалоговой системы оформления первичных документов по заявкам — это «Нормативные документы», «Предметная область» (патентная, непатентная литература, в том числе сведения о базовых объектах и их технико-экономических показателях, знания автора предполагаемого изобретения), «Реквизиты» (данные о предприятии-заявителе, его руководстве, об авторах изобретения). Внешние среды никак не связаны друг с другом.

В первой реализации рассматриваемой системы [1] программы содержат образ всех составляющих внешней среды в виде постоянной части заявок и методической подсказки.

Постоянная часть заявок содержится в виде готовых фрагментов текста заявки, причем в тексте заявки — это отражение среды «Нормативные документы» и «Предметная область», а в остальных документах добавляется отражение среды «Реквизиты». Имеется также файл «Реквизиты авторов». Кроме того, таблица правил формирования (ТПФ) — отражение среды «Нормативные документы» — в случае нахождения по запросу автора (в файле «Реквизиты авторов») его реквизитов представляет их в нужном виде в различных документах. ТПФ позволяет также из некоторых введенных автором исходных данных, например из формулы изобретения, автоматически формировать статику, прототип, недостатки прототипа и некоторые другие части заявки [1].

Все эти внутрисистемные отражения внешней среды были включены в алгоритмы и ПО. Это существенно упростило программы, но такая система не может взаимодействовать с внешней средой, т. е. переналадка, модификация системы в ответ на изменение внешней среды без изменения алгоритма обработки информации и программного обеспечения невозможна.

Опыт эксплуатации этой системы показал, что при частых изменениях во внешних средах (в нормативных документах, например, ежегодно) время жизни такой системы существенно меньше времени творческой активности автора в конкретной предметной области. Кроме того, в предметной области у автора накапливаются знания в результате собственной работы или изучения источников: создаются новые объекты, развивается и изменяется сама предметная область и, следовательно, необходимо изменять математическое обеспечение. Это неудобно, потому что для пользователя частая связь с программистами и патентоведомы затруднительна.

Итак, основной недостаток реализованной диалоговой системы по оформлению заявок — ее консервативность (невозможность взаимодействия ее с внешней средой).

Время жизни диалоговой системы существенно увеличится благодаря накоплению и использованию в ней информации на протяжении всей творческой активности пользователя.

На рис. 1 представлены основные информационные связи системы с каналами входа образов внешних сред.

Части системы, зависимые от внешней среды, должны гибко модифицироваться без разработки дополнительного программного обеспечения, причем желательно при участии самой системы. Поэтому ПО должно быть предметно-независимым и реквизитно-независимым, а зависимым только от самого факта существования первичных документов. Внешние среды должны присутствовать в системе в виде реализующих структур (РС). Реализующие структуры — это фрагменты текста документов с незаполненными участками.

Среда «Нормативные документы» может быть представлена в виде основных реализующих структур (ОРС), пересечение сред «Нормативные документы» и «Предметная область» — в виде вторичных реализующих структур (ВРС), а среда «Реквизиты» — в виде файла «Сервисные знания» (ФСЗ). Таким образом, таблицу правил формирования можно разделить на отдельные части, относящиеся к разным предметным средам. Таблица правил формирования содержит алгоритмы заполнения только частей, производных от введенных пользователем при оформлении частей заявки (заполняются в автоматическом режиме).

Кроме того, в ПО предусмотрены программы, помогающие быстро сменить или модифицировать реализующие структуры в диалоговом или в автоматическом режиме при внешнем обслуживании (при использовании возможностей ВНИИГПЭ).

Изменения ОРС автором заявки не допускаются, поскольку они определяются нормативными документами (их автор менять не вправе). Эта процедура осуществляется только в автоматическом режиме (заменой ОРС).

При желании пользователь может модифицировать ВРС. С точки зрения пользователя, такая модификация даст тот же результат, что и изменение ОРС. Однако изменения во вторичных реализующих структу-

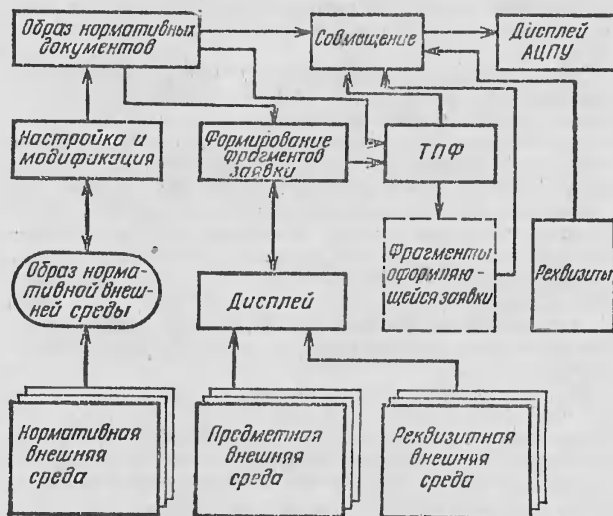


Рис. 1. Основные информационные связи диалоговой системы, взаимодействующей с внешними средами, и каналы входа образов внешних сред

рах система анализирует с точки зрения соответствия среде Нормативные документы», что затруднительно в случае изменения ОРС. В пользу такого разделения говорит и то, что первичные документы с одинаковыми ОРС частично можно обрабатывать в автоматическом режиме. Дополнительное преимущество — пользователь такой системы может совершенно не знать нормативных документов, однако все его заявки будут методически безупречными на протяжении всего времени ее эксплуатации.

В такой системе (рис. 2) широко использован принцип самообучения. Встречая в тексте основной РС указание на какую-либо часть заявки, сначала проверяют наличие этой части в таблице размещения информационно-независимых частей заявки (ИНЧ). В случае успеха система извлекает эту информационно-независимую часть из таблицы и совмещает с ОРС в автоматическом режиме. В противном случае просматривается ТПФ, чтобы определить, является ли эта часть информационно-независимой. Если часть не относится к ИНЧ, то в ТПФ содержатся правила ее формирования из ИНЧ заявки. Эти правила реализуются автоматически, как и совмещение полученного в результате их применения фрагмента текста с ОРС (без запроса пользователя). Если часть относится к ИНЧ, то ТПФ может содержать указание на файл ВРС, необходимых для заполнения данной части. Если нет, то неизвестная часть запрашивается у пользователя. Все ИНЧ заявки, как только они вводятся пользователем, включаются в таблицу размещения ИНЧ и с этого момента не требуют повторного ввода, автоматически совмещаясь с ОРС при всех дальнейших их упоминаниях в ее тексте или текстах других реализующих структур, использующихся при оформлении данной заявки.

Внешние среды отражены в ОРС первичных документов, поставляемых совместно с системой. Они должны по мере необходимости заменяться на ОРС, получаемые от разработчиков. ИНЧ старых заявок пользователей хранятся системой в виде, допускающем повторное использование их или их фрагментов для написания новых заявок, в том числе и под изменившиеся нормативные требования (т. е. под изменившиеся структуры, полученные от разработчика). Это позволяет максимально облегчить труд пользователя при написании заявок по повторяющейся тематике.

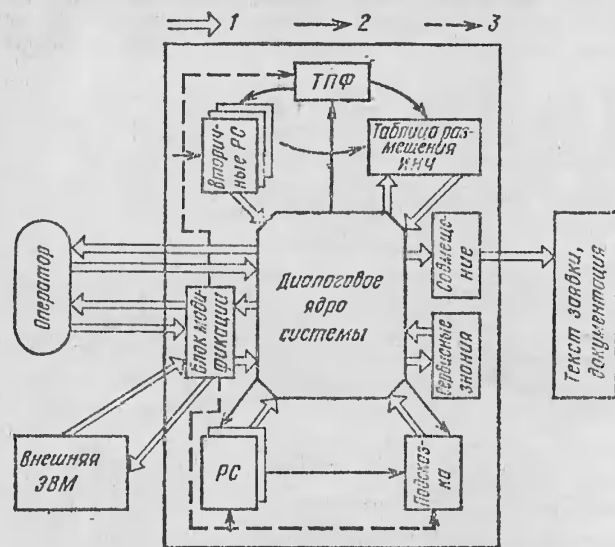


Рис. 2. Вторая реализация диалоговой системы для оформления первичных документов по заявкам:
1 — передача информации и управления, 2 — передача управления, 3 — модификация системы

Система реализована так, что любое изменение ОРС качественно не изменяет диалога и возможности использовать накопившееся представление внешних сред. Меняется лишь текст оформленных документов.
Телефон для справок — 261-11-09 (Москва).

ЛИТЕРАТУРА

1. Сарьян В. К., Александров В. А., Смолич Г. Г., Ефимова Л. Г. Использование реализующих структур для оформления первичных документов по заявкам и формализованная проверка описаний к изобретениям. — Материалы III науч.-техн. конф. «Проблемные вопросы совершенствования системы выявления и защиты советских изобретений в процессе государственной научно-технической экспертизы». — М.: Госкомизобретений, 1986, с. 96—101.
2. Сарьян В. К., Смолич Г. Г. Взаимодействие больших и малых баз данных в автоматизированной системе оформления заявок, ориентированной на индивидуальные пользователей // Микропроцессорные средства и системы. — См. наст. номер. — С. 37—39.
3. Дородницын А. А. Математика и описательные науки (по материалам чл.-кор. АН СССР А. А. Ляпунова). — В кн.: Число и мысль. — М.: Знание, 1977, с. 13—21.

Статья поступила 20 февраля 1987 г.

Уважаемая редакция!

С микроЭВМ «Электроника ДЗ-28» я работаю уже пять лет. За это время накопилось немалое количество программ, а также некоторый опыт использования микроЭВМ разных модификаций. Хотелось бы с помощью Вашего журнала поделиться программами с другими пользователями вычислительных систем ИПГ.

Основные недостатки системы на базе микроЭВМ «Электроника ДЗ-28» — отсутствие накопителей на магнитных дисках (что очень замедляет обмен информацией) и плохое программное обеспечение. В комплект поставки включен пакет статистических программ, которые можно использовать только автономно на микроЭВМ «Электроника ДЗ-28». Этот пакет даже не предусматривает накопления информации и использует всего несколько из 32К байт памяти ЭВМ для работы в режиме калькулятора (хотя можно использовать МЛ). В комплектации завода есть Бейсик-интерпретатор, дающий возможность программировать на алгоритмическом языке, очень удобном для начинающего программиста. Но это не компилятор, и поэтому быстродействие ЭВМ уменьшается почти в 15 раз. Это неудобство исключается языком программирования ОС ВТ МХТИ, созданным в МХТИ на основе элементов нескольких языков программирования (он значительно сложнее, чем Бейсик 3А).

Мною создан комплекс программ (на языке Бейсик 3А и ОС ВТ МХТИ) для накопления, хранения, обработки и отображения данных. Наиболее эффективна программа для многофакторного корреляционного анализа. Программа дает возможности: ввода данных с дисплея; просмотра данных (печать); записи (чтения) данных на МЛ; коррекции данных; формирования уравнения; счета коэффициентов управления; статистического анализа остатков модели; любых преобразований данных по столбцам; графического отображения результатов.

Разработаны программа описания экспериментальной кривой, непрямолинейной в отношении коэффициентов (когда нельзя использовать метод наименьших квадратов); программа обработки и печати текстовой информации.

Лекис Витаутас Станиславович

Адрес для справок: 234324, Каунас-Академия, ЛитСХА, каф. лесоустройства. Тел. 29-65-26.

Что дает такая система в целом для изобретательства?

Для заявителя она прежде всего сокращает на 30...70% время подготовки и оформления первичных документов по заявкам, так как в систему заложены все действующие нормативно-методические материалы по написанию текста заявки, и в режиме интерактивного диалога с ЭВМ автор-пользователь отвечает на вопросы ЭВМ, не выходя за рамки знакомой ему предметной области. Заложенные в системе знания об информационной избыточности текста и документов по заявке позволяют пользователю накапливать в процессе эксплуатации опыт в реквизитной, предметной и нормативно-методической областях. Знания в системе при необходимости легко изменить. Документы по заявке, введенные в ЭВМ, позволяют формировать информацию в базу данных АСУ, действующей на предприятии.

Для значительного числа потенциальных изобретателей упрощаются и

ускоряются подготовка и оформление документов по заявкам. Экспертные совещания благодаря аппаратуре «Спринт» возможны без выезда изобретателей и патентоведов во ВНИИГПЭ.

Во ВНИИГПЭ при использовании БДЭН сокращается до нуля возврат заявок на стадии предэкспертизы (сейчас он составляет 12...15 тыс. заявок в год); отпадает необходимость в приеме тысяч командированных, приезжающих ежегодно на экспертные совещания; появляется возможность готовить в автоматизированном режиме решения экспертизы, описания на изобретения и рефераты (при существующих потоках — 80 тыс. в год), корректировать и передавать их по каналам связи или на машинных носителях, а также проводить экспертные совещания (без выезда заявителей во ВНИИГПЭ) и обмен корреспонденцией с заявителями по каналам связи.

Успешно испытаны фрагменты автоматизированной сети по созданию за-

явочной документации в ее подготовке на бумажной ленте и гибком диске, а также в памяти ЭВМ с дальнейшей передачей ее ВНИИГПЭ для экспертизы в соответствии с действующей технологией. Кроме того, документация передавалась из ЭВМ во ВНИИГПЭ и по телефонной линии связи с вводом в банк документов. Как показали итоги опытной эксплуатации, заявитель избавляется от необходимости глубоко изучать инструктивно-методические материалы по созданию заявочной документации. Ему дается инструмент, в основу которого заложены все основные требования по созданию заявки в режиме дружеской подсказки. Заявитель только отвечает на вопросы. Необходимость в постоянной консультации высококвалифицированного работника патентной службы отпадает.

Телефон для справок: 261-11-09, 240-30-74 (Москва).

Статья поступила 10 октября 1986 г.

МАШИННАЯ ГРАФИКА

ОТ РЕДАКЦИИ. Успехи машинной графики самым непосредственным образом связаны с развитием микроэлектроники и микропроцессорной техники. С одной стороны — широкое распространение персональных компьютеров, рассчитанных на пользователя без специальной подготовки, потребовало удобных средств визуализации информации. С другой стороны — микропроцессоры позволяют повысить «квалификацию» алфавитно-цифрового дисплея, дополнив его ограниченными графическими возможностями: превратить обычный телевизор в цветной расстрельный дисплей; заменить в векторном дисплее традиционный дисплейный процессор, расширяя при этом его функции. Микропроцессорная техника стимулирует создание новых архитектурных дисплеев, в которых используются комбинированные и мат-

ричные принципы обработки графической информации в реальном масштабе времени.

При массовом применении средств машинной графики необходима стандартизация интерфейсов. Существует международный стандарт GKS, определяющий базовые графические средства и связь с языками программирования. Ведется работа по стандартизации операций с окнами.

Базовое графическое обеспечение служит основой для создания разного рода универсальных и специализированных графических редакторов, которые в свою очередь используются в системах автоматизации проектирования, научных исследований и т. д.

В этом номере и в последующих журнал будет знакомить читателей с работами по машинной графике.

УДК 681.3.022

А. П. Демин, Г. И. Харитонов

УСТРОЙСТВО ОТОБРАЖЕНИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ НА АЛФАВИТНО-ЦИФРОВОМ ДИСПЛЕЕ

По экономическим и эксплуатационным соображениям целесообразно использовать один дисплей для отображения цифровой и графической информации. В графических дисплеях принято поэлементное управление изображением*, что требует большего объема памяти и

* Козак А. А., Сорока С. И. Устройства отображения информации микропроцессорных средств вычислительной техники // Микропроцессорные средства и системы, — 1984. — № 3. — С. 27—32.

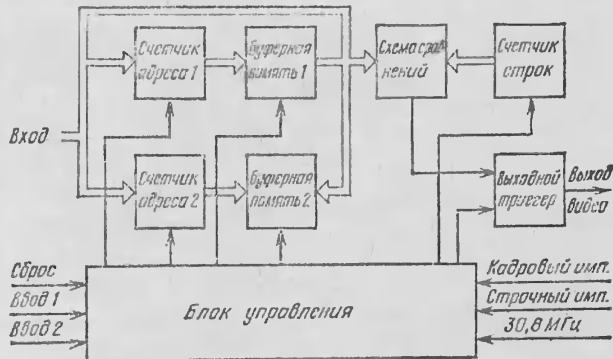


Рис. 1. Структурная схема устройства отображения

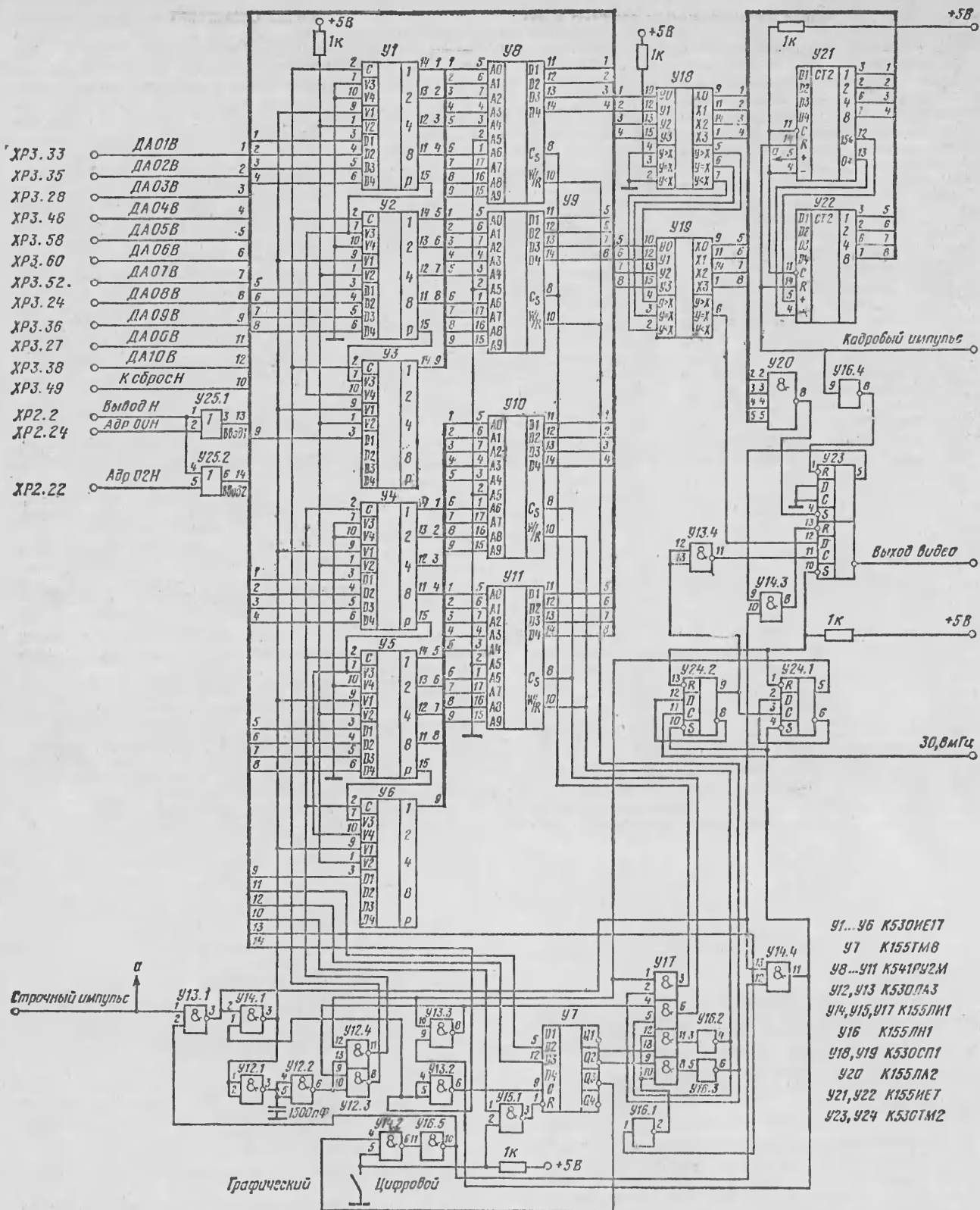


Рис. 2. Принципиальная схема устройства отображения

средств быстрого обмена данными между памятью и экраном.

Часто требуется отображение одного графика. С помощью устройства (рис. 1) можно организовать отображение графической информации на алфавитно-цифровом дисплее 15ИЭ-00-013. Размер графического изображения выбран 256×800 элементов. Тактовое время растровых точек при этом составляет 65 нс. Время цикла считывания применяемых в буферной памяти микросхем серии К541 равно 120 нс, поэтому буферная память организована в виде двух блоков, работающих попеременно через один такт. Информация, поступающая из буферной памяти, сравнивается с номером строки, и при равенстве кодов сигнал с выходного триггера высвечивает элемент изображения на мониторе. Устройство имеет два режима: ввод информации в буферную память и вывод информации на монитор. Блок управления определяет режим устройства.

Режим работы определяется переключателем или триггером D3 микросхемы У7 (рис. 2). Когда переключатель находится в положении «Графический» или в D3 занесена «Лог. 1», устройство выводит информацию из буферной памяти на монитор. При положении «Цифровой» или в триггер D3 занесен «Лог. 0» устройство находится в режиме ввода информации в буферную память. Триггер D3 сбрасывается переключателем либо импульсом «Сброс».

По сигналу «Ввод 2» информация со входа ДА01... ДА010 передается в счетчик адреса 1 и 2, а также в триггеры D2, D3 микросхемы У7. Триггер D2 выбирает буферную память 1 или 2, информация в которую записывается по сигналу «Ввод 1». Для сохранения служебной строки при графическом отображении растр сдвигается на 29 строк с помощью дешифратора У20. При этом происходит излом растра (рис. 3), что необходимо учитывать при вводе информации в буферную память. Растр начинает формироваться на мониторе,

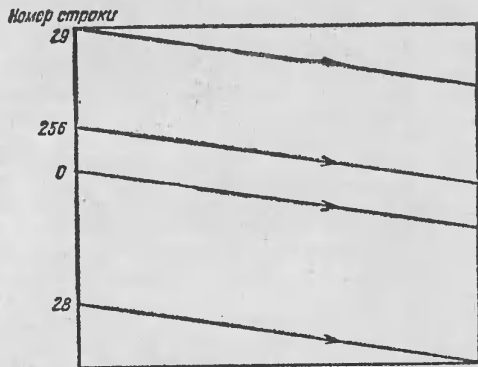


Рис. 3. Растр, формируемый при графическом отображении

когда счетчик строк (микросхемы У21, У22) имеет значение 29, после прихода кадрового синхронизирующего импульса (рис. 4). Для установки адреса первой ячейки буферной памяти необходимо в счетчики адресов занести код 777, а триггер D2 установить в положение «Лог. 0». Адрес второй ячейки соответствует «Лог. 1» в триггере D2. Следующие два адреса ячеек буферной памяти соответствуют коду 000 в счетчиках адресов и определяются триггером D2. Адреса ячеек буферной памяти задают, наращивая код счетчиков адресов на единицу.

Разработанное устройство используется для работы с микроЭВМ «Электроника 60», подключается с помощью стандартного интерфейса И5 и расположено на его свободной половине. На интерфейсе И5 установлены адреса: 177570 для ввода кода в триггеры D2, D3 и счетчики адресов; 177572 для ввода информации в буфер-

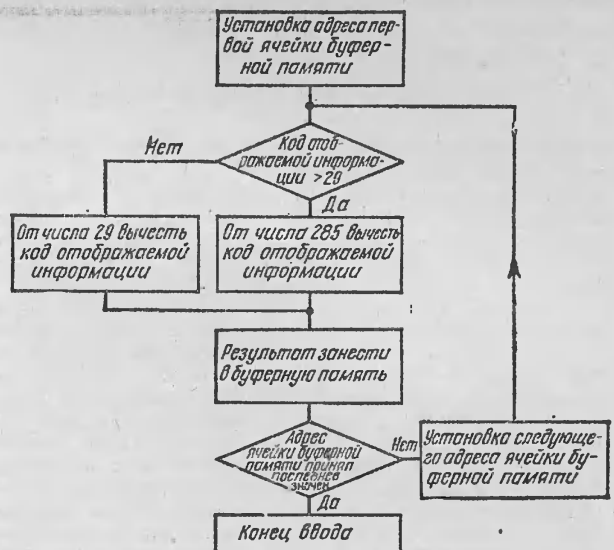


Рис. 4. Блок-схема занесения информации в буферную память

ную память. Сигнал «Ввод 1» формируется при обращении микроЭВМ по адресу 177570, «Ввод 2» по адресу 177572.

Строчный и кадровый синхронимпульсы подаются с платы генератора символов дисплея. Тактирующий сигнал 30,8 МГц поступает с микропрограммного устройства через высокочастотные соединители СР-50. Видеовыход подключен через соединители СР-50 к выводу 1 микросхемы D11, платы генератора символов. Соединение между выводами 1,2,4,5 и 9,10,12,13 микросхемы D11 при этом разрывается.

Ассемблер макро-11

Программа пересылки информации из микроЭВМ в буферную область дисплея

В R2 задается начальный адрес выводимого массива

```

MOV # 1776, R4
MOV # 177570, R0
MOV # 177572, R1
M3: MOV R4, (R0)
    MOV (R2)+, R5
    CMP # 35, R5
    BHI M1
    MOV # 435, R3
    SUB R5, R3
    BR M2
M1: MOV # 35, R3
    SUB R5, R3
M2: MOV R3, (R1)
    INC R4
    CMP # 2775, R4
    BNE M3
    HALT
    
```

Программа, обеспечивающая ввод отображаемой информации в устройство, размещается, начиная с ячейки по адресу 1000.

Телефон для справок — 441-95-66, Киев.

Статья поступила 10 февраля 1986 г.

УСТРОЙСТВО ИНДИКАЦИИ

Для вывода алфавитно-цифровой и (или) графической информации (например, на этапах отладки и ремонта) создано устройство, позволяющее использовать промышленные осциллографы различных типов. Устройство можно использовать в любых микропроцессорных системах с магистралью (каналом), соответствующей «Единому каналу» микроЭВМ «Электроника 60».

Устройство вырабатывает пилообразные напряжения, подаваемые на вход «Y» и «X» осциллографа. Напряжение на входе «Y» имеет отрицательную полярность, его период (~20 мс) превышает период напряжения на входе «X» в 64 раза. Формируется растр в 64 строки, начинающийся с левого верхнего угла экрана осциллографа. Каждая строка разбивается на 64 равных отрезка. Каждый полуотрезок может быть подсвечен импульсным напряжением, подаваемым на вход «Z» осциллографа. Визуально подсвеченный отрезок индицируется в виде точки (максимальное число на экране — 4096). Наличие или отсутствие каждой конкретной точки определяет конкретный бит (хранится в ОЗУ устройства). Для ЭВМ микросхема ОЗУ программно доступна по записи. Разработанные программы формируют с помощью ЭВМ точечные изображения цифр, букв, символов, графиков и т. д. Изображение индицируется устройством без участия ЭВМ.

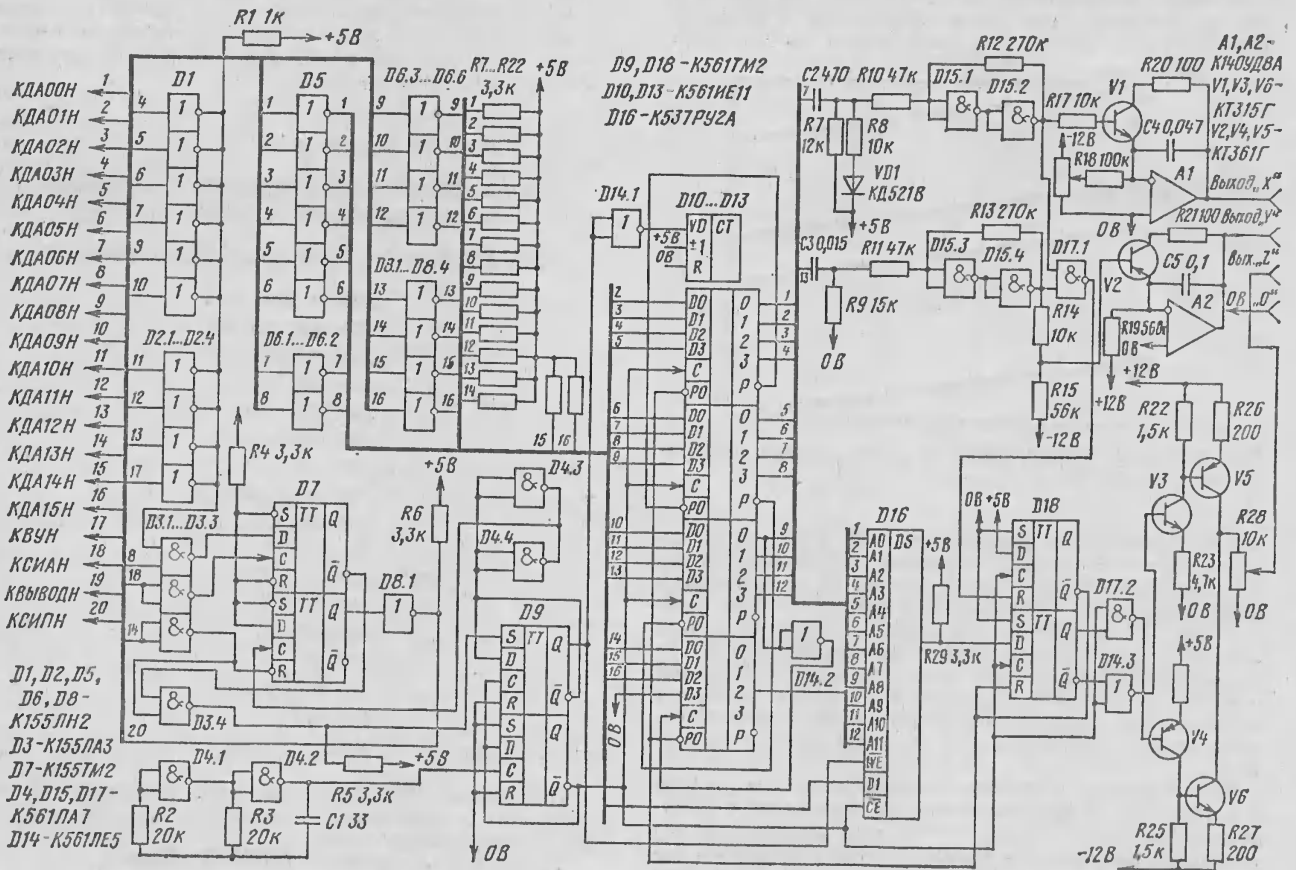
Центральный элемент схемы (см. рисунок) — микросхема ОЗУ К537РУ2А (4К×1). Основное время микросхема работает в режиме «Считывание». Счетчик D10, D11, D12 периодически перебирает все адреса. Выход-

ная информация микросхемы фиксируется триггером D18. Он же управляет модулятором напряжения «Z» (на элементах D17.2, D14.3 и транзисторах V3, V4, V5, V6). Если на выходе триггера — «Лог, 0», то напряжение на «Z» равно нулю, если — «Лог, 1», то на «Z» — знакопеременный меандр с частотой, равной частоте тактового генератора (на элементах D4.1, D4.2, D9.2). Уровень подсвета любой точки не зависит от вида всего изображения, так как сигнал на «Z» не имеет постоянной составляющей. Амплитуда данного сигнала регулируется подстроечным резистором R28.

Пилообразные напряжения для входов «X» и «Y» вырабатывают интеграторы на операционных усилителях A1, A2. Оба интегратора имеют транзисторные ключи сброса V1, V2.

Интегратор «X» сбрасывается по отрицательному перепаду 6-го разряда счетчика адреса. Длительность прямого хода по «X» — 64 такта. Дифференцирующая цепочка C2, R7 определяет длительность импульса сброса, а элементы D15.1, D15.2 формируют его фронт и спад.

Интегратор «Y» сбрасывается после индикации 64 строк растра, т. е. всего объема ОЗУ D16. Импульс сброса вырабатывается по отрицательному перепаду 3-го разряда счетчика D13. Данный счетчик облегчает перевод строки при выводе на индикатор текстовой информации. При синфазной работе счетчика D13, программно доступного для ЭВМ по записи, и счетчика старших разрядов адреса D12 импульс сброса интегратора «Y» будет вырабатываться после отрицательного перепада старшего разряда адреса ОЗУ. При этом информация, записанная по старшим адресам ОЗУ, индицируется внизу экрана. Запись смещения на одну дискрету в счетчик D13 вызовет появление импульса сброса



Принципиальная схема устройства индикации

на 512 тактов раньше. При этом все изображение циклически сместится вверх на 8 строк раstra. Это соответствует одной строке текста (если размер алфавитно-цифровых символов по высоте не превышает семи точек). Дальнейшая программная очистка одной восьмой части объема ОЗУ (соответствует индикации восьми нижних строк раstra) завершит цикл перевода строки. Этот способ не требует условия считывания или дублирования ранее выведенной в D16 информации.

На время обратного хода луча по «Х» и «У» счетчики адреса затормаживаются сигналом триггера D18.2.

В обмене по каналу микроЭВМ устройство участвует только в цикле «Вывод». Ему присвоен адрес 177570, обращение по которому выделяется элементами D1, D2.1... D2.4 и фиксируется триггером D7.1 по сигналу КСИАН.

Временные сопряжения осуществляют триггеры D9.1 и D7.2. Первый из них переводит счетчики адреса в режим параллельной записи и формирует сигнал «Запись» для микросхемы ОЗУ D16. Его переключение происходит

после окончания предыдущего цикла считывания ОЗУ. Это обеспечивает установку адреса, данных и сигнала «Запись» на входах микросхемы ОЗУ в момент, когда сигнал на входе «Выбор кристалла» — пассивный. По активному уровню этого сигнала (низкому) информация записывается. По окончании записи триггером D7.1 и элементом D8.1 вырабатывается сигнал ответа КСИПН. Максимальная задержка ответа не превышает двух тактов задающего генератора (период такта ~4 мкс).

Программа данного устройства индицирует символы согласно кодам КОИ-7₂. В микросхеме ОЗУ по нужному адресу информация записывается одним управляющим словом: разряды D00 — записываемая информация, D01...D12 — адрес, D13...D15 — номер нижней строки текста.

Адрес для справок: 330600, г. Запорожье, ГСП-187, ул. Рельефная, д. 18, ВНИИпреобразователь.

Статья поступила 4 октября 1986 г.

УДК 681.326—181.4

П. А. Семенов, А. М. Процак, В. П. Егоров

ЦВЕТНАЯ ГРАФИКА В МИКРОЭВМ «ЭЛЕКТРОНИКА 60» И «ЭЛЕКТРОНИКА НЦ-80»

Применение цветных графических дисплеев значительно расширяет функциональные возможности микроЭВМ, так как позволяет выводить информацию в форме, удобной для восприятия [1]. Графическая программная библиотека позволяет просто и эффективно использовать графические возможности микроЭВМ совместно с языками высокого уровня и унифицировать графический язык для различных устройств.

Рассмотрим аппаратно-программные средства построения контроллеров цветной графики (CGC) с различными характеристиками для микроЭВМ «Электроника 60» и «Электроника НЦ-80». Совместно с графическим ПО на языках Паскаль и ассемблер эти контроллеры представляют собой законченные мощные графические средства для применения в цифровой обработке сигналов (при автоматизированном управлении, проектировании и т. д.).

Базовый вариант — 16-цветный CGC (256×256 элементов изображения), на основе которого (рис. 1) дальнейшей модификацией можно построить контроллеры с более высоким разрешением и градациями яркости. Контроллер занимает 13 портов ввода-вывода и

два порта обращения к видеопамяти VRAM. Обращение к VRAM, как к порту ввода-вывода (последовательный доступ), экономит адресное пространство микроЭВМ и позволяет использовать основную память под большие массивы данных, однако требует предварительной загрузки регистра адреса VRAM. Включение VRAM в адресное пространство памяти микроЭВМ (произвольный доступ) снижает время обмена данными (примерно 3 мкс на каждое обращение).

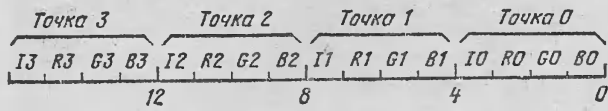
Основные узлы контроллера — блок сопряжения с системным каналом микроЭВМ «Электроника 60» SYSIF, двухпортовый контроллер регенерации изображения DPRC, блок видеопамяти VRAMU, блок синхронизации SYNCHU и декодер VCODER.

Блок SISIF, осуществляющий обмен информацией между CGC и микроЭВМ, включает в себя двунаправленный передатчик адреса данных FAD, ПЗУ опознавания запроса APROM, блок декодирования адреса портов ввода-вывода I/OD, входящих в контроллер, схему формирования управляющих сигналов и сигналов запроса VRAM. Перепрограммированием APROM адреса VRAM можно включить в пространство памяти микроЭВМ.

В состав блока VRAMU, предназначенного для хранения кодов элементов изображения одного кадра, входят видеопамять VRAM, регистр адреса видеопамяти VRAMARG, мультиплексор адреса видеопамяти VRAMAMUX, регистр считываемых данных VRAMDRG и сдвиговый регистр регенерации SRG. Слово видеопамяти (рис. 2) включает в себя четыре элемента изображения, каждый из которых кодируется тремя разря-

→ К стр. 48

Рис. 1. Функциональная схема 16-цветного контроллера графики с разрешением 256×256 элементов изображения. CGC — контроллер цветной графики; SYSIF — блок сопряжения с системным каналом микроЭВМ; DPRC — двухпортовый контроллер регенерации изображения; VRAMU — блок видеопамяти; SYNCHU — блок синхронизации; VCODER — декодер; FAD, F — формирователи; ARG — регистр адреса; APROM — ПЗУ опознавания запроса; I/OD — блок декодирования адреса портов; T1, T2, T3 — программируемые таймеры; S — одновибраторы; VRAMARG — регистр адреса видеопамяти; VRAMAMUX — мультиплексор адреса видеопамяти; VRAMDRG — регистр данных видеопамяти; VRAM — видеопамять; SRG — сдвиговый регистр регенерации; LC — строчный счетчик; FC — кадровый счетчик; EOL — импульс конца строки; LBP — строчный гасящий импульс; LSPB — задержка строчного синхронимпульса; LSYN — строчный синхронимпульс; EOF — импульс конца кадра; FBP — кадровый гасящий импульс; FSPB — задержка кадрового синхронимпульса; FSYN — кадровый синхронимпульс; Synch — сигнал синхронизации; Video Blank — сигнал управления для декодера; MARK-X, MARK-Y — координаты маркера; Mark EN, Mark Blink — сигналы управления маркером; Mark-AX, Mark-AY — размеры маркера; Marker, Video EN — сигналы управления декодером; IORQ — запрос портов ввода-вывода; VRAMRQ — запрос видеопамяти; MARKER TIMER1, TIMER2, TIMER3, VRAM READ, VRAMADR — сигналы обращения к портам ввода-вывода и видеопамяти; RD/WR — ввод-вывод; VRAMSTB — обращение процессора к видеопамяти; RAS, CAS, W/R — сигналы управления видеопамятью; SRGLOAD — загрузка сдвигового регистра регенерации; VRAMDATASTB — загрузка регистра левых видеопамяти; VRAMACK — подтверждение обработки запроса в DPRC; Ввод, Вывод, ВУ, СИАС, СИПС — системные сигналы микроЭВМ.



Код цвета				Цвет
I	R	G	B	
0	0	0	0	Черный
0	0	0	1	Синий
0	0	1	0	Зеленый
0	0	1	1	Пурпурный
0	1	0	0	Красный
0	1	0	1	Фиолетовый
0	1	1	0	Желтый
0	1	1	1	Белый
1	0	0	0	Серый
1	0	0	1	Яркий синий
1	0	1	0	Яркий зеленый
1	0	1	1	Яркий пурпурный
1	1	0	0	Яркий красный
1	1	0	1	Яркий фиолетовый
1	1	1	0	Яркий желтый
1	1	1	1	Яркий белый

Рис. 2. Формат слова видеопамати

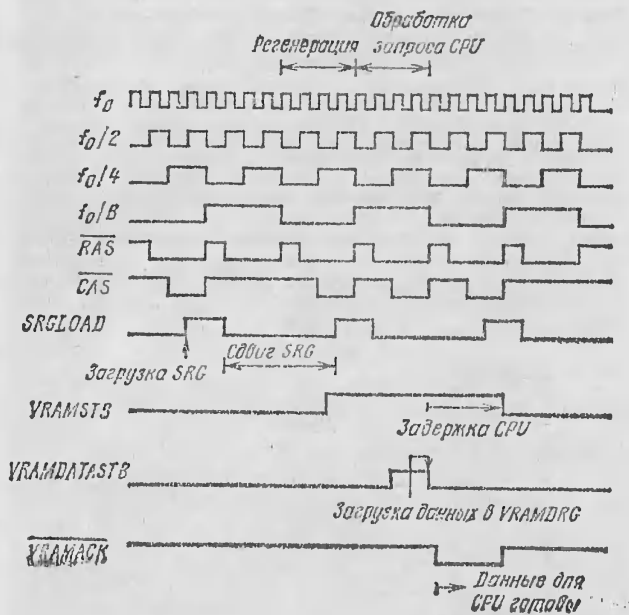


Рис. 3. Временные диаграммы работы двухпортового контроллера регенерации изображения

f_0 — опорная частота; RAS, CAS — сигналы загрузки адреса в видеопамати; SRGLOAD — загрузка сдвигового регистра регенерации; VRAMSTB — обращение процессора к видеопамати; VRAMDATASTB — загрузка регистра данных видеопамати; VRAMACK — подтверждение обработки запроса в DPRC

рует счетчик LC/FC. За цикл VRAM производится три сдвига в SRG. Одновременно с регенерацией изображения производится регенерация динамической памяти.

При обработке запроса процессора, сопровождающегося наличием сигналов VRAMRQ и RDVWR, адрес VRAM формируется с выхода регистра VRAMARG, который должен быть предварительно загружен при

последовательном способе доступа. Тип запроса определяется наличием сигнала RD (ввод^с), который управляет формированием W/R для VRAM. В случае считывания данных из буферного регистра VRAMDRG он загружается сигналом VRAMDATASTB по окончании CAS. При распознавании обращения к VRAM интерфейсный блок SYSIF блокирует выдачу системного сигнала СИП^с синхронно с сигналами Ввод^с/Вывод^с и переключается на ожидание подтверждения обработки запроса в DPRC (сигнал VRAMACK), который также формируется при сбросе CAS.

Основные элементы блока синхронизации SYNCHU (рис. 4), формирующего сигнал синхронизации Synch для TV-монитора, сигнал управления Video Blank для видеокодера и сигналы сброса LC/FC в DPRC — это программируемые таймеры KP580BI53 [2]. Длительность строк и кадров задается сигналами: конец строки EOL (T1#0) и конец кадра EOF (T2#0), а информационные длины строк и кадров определяются строчным LBP (T1#1) и кадровым FBP (T2#1) гасящими импульсами. Сигнал синхронизации Synch представляет собой последовательность строчных LSYN и кадровых FSYN синхронимпульсов, сдвинутых относительно передних фронтов гасящих импульсов. Длительности синхронимпульсов задаются одновибраторами, а их задержки формируются таймерами T1#2 и T2#2.

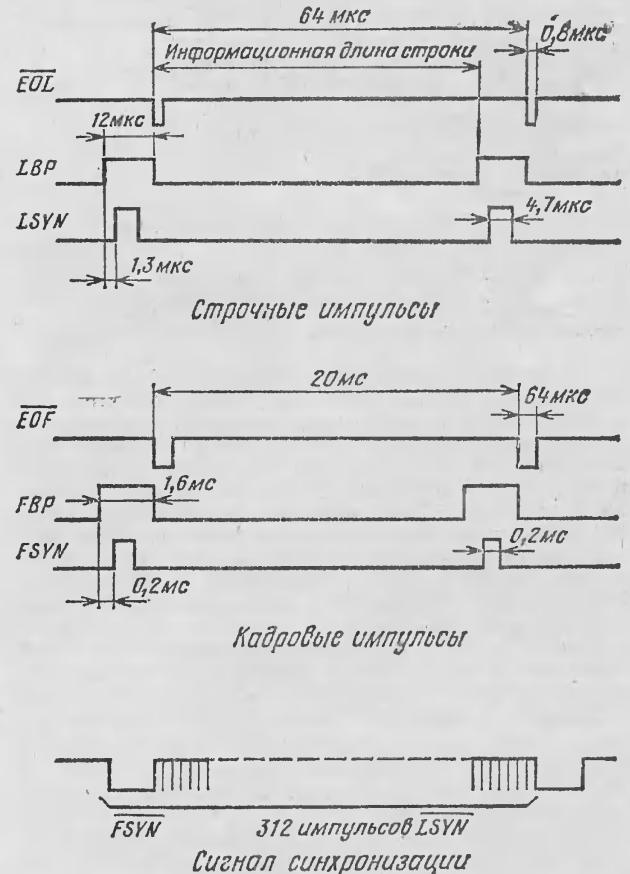


Рис. 4. Форма сигнала синхронизации

EOL — импульс конца строки; LBP — строчный гасящий импульс; LSYN — строчный синхронимпульс; EOF — импульс конца кадра; FBP — кадровый гасящий импульс; FSYN — кадровый синхронимпульс; Synch — сигнал синхронизации

Для удобства работы с дисплеем в алфавитно-цифровом режиме в состав SYNCHU введен блок формирования маркера: программно-перемещаемого по экрану и мигающего (3 Гц) изображения прямоугольника с аппаратно-изменяемыми геометрическими размерами MARK-ΔX и MARK-ΔY, задаваемыми одновибраторами и программно-изменяемыми таймером T3 координатами MARK-X и MARK-Y.

Формирование RGB-сигнала для TV-монитора происходит в видеокоде, вариант схемы которого для формирования шестнадцати цветов в соответствии с рис. 2 приведен на рис. 5. Цифровым мультиплексором MUX управляют импульсы MARK и Video Blank, из блока SYNCHU разрешающие формирование видеосигналов для элемента изображения или маркера, а также выключающие изображение на время гасящих импульсов. Цвет маркера задается программно в регистре MARKCOLORRG. Уровень выходного сигнала формирователей определяется битом I каждого элемента изображения.

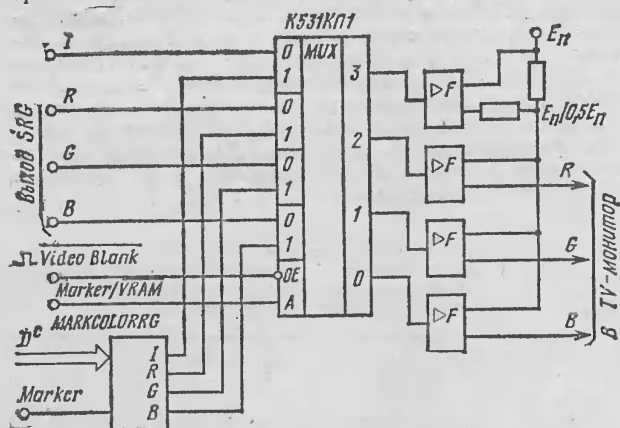


Рис. 5. Структурная схема видеокодера

MARKCOLORRG — регистр цвета маркера; MUX — цифровой мультиплексор; F — формирователи видеосигнала; Video Blank, Marker/VRAM — сигналы управления мультиплексором; MARKER — обращение к регистру цвета маркера; SRG — сдвиговый регистр регенерации изображения

Повысить разрешающую способность дисплея при сохранении быстродействия VRAM можно кодированием в слове VRAM большего числа элементов изображения. Это уменьшит количество отображаемых дисплеем цветов. Например, при разрешении 512×256 и кодировании в 16-разрядном слове VRAM восьми элементов изображения позволяет воспроизводить только четыре цвета, что значительно ограничит возможности графики. Для сохранения количества воспроизводимых цветов необходимо увеличить длину слова VRAM до 32 разрядов. Это потребует от процессора двух циклов для обмена данными с VRAM (рис. 6).

Данные в VRAM записываются в два этапа. После адресации к ячейке VRAM процессор записывает данные в младшие 16 разрядов VRAML, а следующей командой — в старшие VRAMH. Это обеспечивается избирательной подачей импульсов CAS. При регенерации импульсы CAS одновременно подаются в VRAMH и VRAML, разрешая считывание 32 бит VRAM в SRG и далее в видеокодер.

Фиксированный набор воспроизводимых цветов и отсутствие плавного изменения яркости изображения ограничивают применение рассмотренных выше CGC. Более широкими функциональными возможностями обладает контроллер с программируемым цветовоспроизведением (рис. 7). Выходной RGB-сигнал формируется цифро-аналоговыми преобразователями DACR, DACG, DACB (например, К118ПА1А или ЦАП-10), разряд-

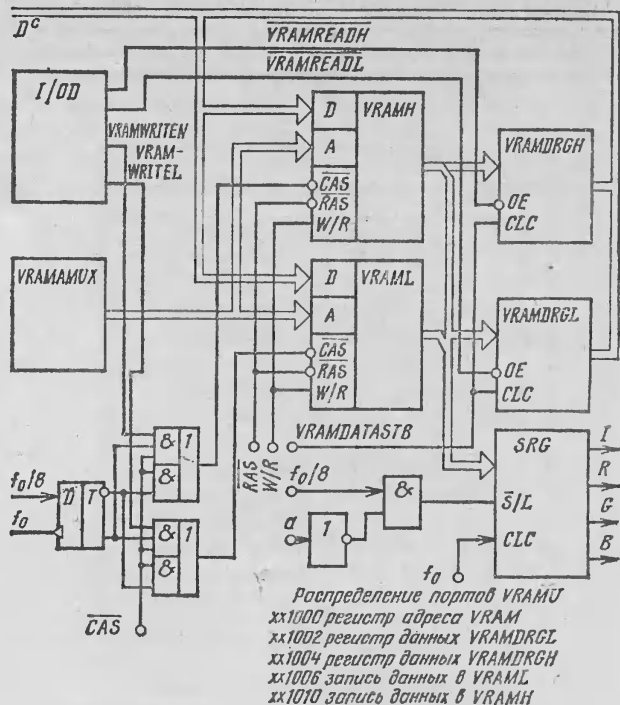


Рис. 6. Структурная схема блока видеопамати 16-цветного контроллера цветной графики (512×256 элементов изображения)

I/OD — блок декодирования адресов портов; VRAMMUX — мультиплексор адреса видеопамати; VRAML — блок младших разрядов видеопамати; VRAMH — блок старших разрядов видеопамати; VRAMDRGL, VRAMDRGH — регистры данных для младших и старших разрядов слова видеопамати; SRG — сдвиговый регистр регенерации изображения; RAS, CAS, W/R — сигналы управления видеопаматью; VRAMWRITEL — запись данных в младшие разряды видеопамати; VRAMWRITELH — запись данных в старшие разряды видеопамати; VRAMREADL — чтение данных из младших разрядов видеопамати; VRAMREADH — чтение данных из старших разрядов видеопамати; VRAMDATASTB — загрузка регистров данных видеопамати

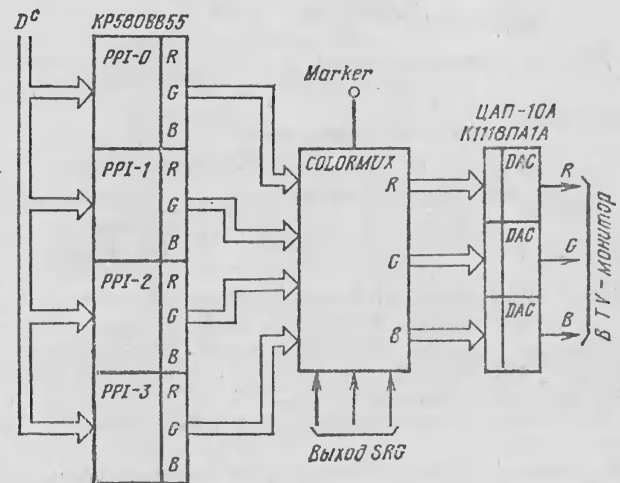


Рис. 7. Структурная схема видеокодера контроллера с программируемым цветовоспроизведением

PPI — программируемый параллельный интерфейс; DAC — цифро-аналоговые преобразователи; COLORMUX — мультиплексор цвета

ность которых определяет количество градаций яркости цветов. Набор воспроизводимых цветов (палетта) задается программно записью разложения RGB выбранного цвета в выходные регистры программируемых параллельных интерфейсов PPI. В зависимости от кода элемента изображения, который представляет собой порядковый номер цвета выбранной палетты, мультиплексор цвета COLORMUX передает на входы DACR, DACG, DACB код RGB разложения соответствующего цвета. Данный контроллер с программируемым цветовоспроизведением может найти широкое применение при обработке полутонных изображений и т. п.

Функционирование дисплея обеспечивается графической библиотекой программ GRLIB, которая может быть включена в основную программу на этапе ее компоновки в операционной системе (ОС), совместимой с RT-11. GRLIB, написанная на языке Паскаль и ассемблер, реализует графический монитор GDM [3] высокого уровня со структурой, аналогичной графической версии языка BASICAGL и пакета HP-ISPP [4]. При этом в начало текстового варианта исходной программы (на языке Паскаль) должен быть включен вспомогательный файл с перечнем процедур GRLIB с указанным типом EXTERNAL.

Графический монитор (GDM) включает в себя команды, условно разделенные на три группы.

1. Команды управления режимом:

GRAPHICS — перевод дисплея в графический режим с начальной установкой параметров и стиранием видеопамати;

EXITGRAPHICS — выход из графического режима;

PLOTTERIS — назначение графического устройства;

GCLEAR — стирание видеопамати;

ZCLEAR (X1, X2, Y1, Y2) — стирание зоны видеопамати;

GSTORE (file name) — запись содержимого видеопамати на внешнее устройство;

GLOAD (file name) — загрузка видеопамати с внешнего устройства;

ZONECOLOR (n) — выделение зоны LOCATE цветом COLOR;

PALETTE (m) — назначение палетты цветов изображения;

COLOR (n) — выбор цвета элемента изображения.

2. Команды построения графических структур:

LIMIT (X1, X2, Y1, Y2) — указание на экране зоны, ограничивающей рабочее поле для вывода информации;

LOCATE (X1, X2, Y1, Y2) — выделение на экране зоны для построения графической информации;

FRAME — выделение границ зоны LOCATE на экране дисплея выбранным цветом COLOR;

SCALE (X1, X2, Y1, Y2) — наложение масштаба пользователя на зону LOCATE;

SETUU — назначение системы единиц пользователя;

SETGU — назначение системы единиц графического устройства;

LINETYPE (n) — выбор типа линии;

DRAW (X1, Y1, X2, Y2) — проведение линии от точки (X1, Y1) к точке (X2, Y2);

MOVE (X, Y) — перевод графического курсора в точку с координатами (X, Y);

RMOVE (dX, dY) — перевод графического курсора из текущего положения (X, Y) в точку (X+dX, Y+dY);

PLOT (X, Y) — проведение линии от текущего положения графического курсора до точки (X, Y);

RLOT (dX, dY) — проведение линии от текущего положения графического курсора к точке (X+dX, Y+dY);

SPIRAL (X0, Y0, α 0, α 1, R0, R1) — построение спирали с центром в точке (X0, Y0), начальным радиусом

R0, образующим с осью X угол α 0, и конечным радиусом R1, образующим с осью X угол α 1;

RSPIRAL (R0, R1, α 0, α 1) — построение спирали с текущей позиции графического курсора с начальным радиусом R0 с направлением на центр α 0 до точки с радиусом R1 и углом α 1;

CIRCLE (X0, Y0, R, α 0, α 1) — построение дуги окружности с центром в точке (X0, Y0) радиусом R, образующим с осью X начальный угол α 0 и конечный угол α 1;

RCIRCLE (R, α 0, α 1) — построение дуги окружности с текущего положения графического курсора радиусом R и начальным направлением на центр α 0 до точки с радиусом R и углом α 1;

LINE (R: pixelrecord) — построение кусочно-линейной кривой с узлами в точках {(X0, Y0), (X1, Y1), ..., (XN, YN)};

CURVE (R: pixelrecord) — построение интерполирующей кривой n+1-го порядка с узловыми точками {(X0, Y0), (X1, Y1), ..., (XN, YN)};

AXES (X0, Y0, dX, dY, NX, NY) — построение координатных осей X и Y с центром в точке (X0, Y0), с шагом dX/dY нанесения мелких делений на ось X/Y и интервалами Nx/Ny нанесения крупных делений на соответствующие оси;

GRID (X0, Y0, dX, dY) — нанесение на зону LOCATE координатной сетки пользователя с центром в точке (X0, Y0) и шагом dX/dY по оси X/Y;

GRADIUS — измерение радиуса R проводится в единицах графического устройства;

XRADIUS — масштаб R равен масштабу по оси X;

YRADIUS — масштаб R равен масштабу по оси Y;

3. Команды вывода текстовых сообщений:

TEXTDIR (n) — установка ориентации текста;

CSIZE (n) — выбор размера графического шрифта;

CSLANT — установка наклона 45° графического шрифта;

CLANTOFF — отмена действия команды CSLANT.

Команды первой группы управляют режимами работы дисплея, производят начальную установку параметров процедур LIMIT, LOCATE, SCALE, CSIZE, TEXTDIR, LINETYPE и обмен между видеопаматью и внешними запоминающими устройствами для хранения и восстановления графических изображений. Процедуры PALETTE и COLOR непосредственно не производят никаких операций с дисплеем, а служат для предварительного запоминания и выбора кода цвета с целью его последующей загрузки в видеопамать операторами построения графических структур.

Графическое изображение формируется командами второй группы, синтезирующими любую графическую структуру в единицах измерения формата экрана (256×256, 512×256 и т. д.) и в масштабе пользователя. Для многооконного представления информации на экране в состав GDM введены команды LIMIT (ограничивает текущую рабочую зону с соответствующим переносом начала координат) и LOCATE (ограничивает зону графического изображения, но не влияет на вывод текстовой информации). На зону LOCATE (визуально обозначается командами FRAME и ZONECOLOR) может быть наложен масштаб пользователя командой SCALE. Масштабы зоны LOCATE (или всего изображения) между единицами формата экрана и пользователя оперативно переключаются командой SETGU (или SETUU).

Основа команд синтеза графических изображений — процедура DRAW (рис. 8), строящая отрезок прямой линии между точками с координатами (X1, Y1) и (X2, Y2). Определение типа и необходимое преобразование масштаба — автоматические: считывается внутренняя переменная unittype библиотеки GRLIB (устанавливаемая командами SETUU, SETGU и SCALE).

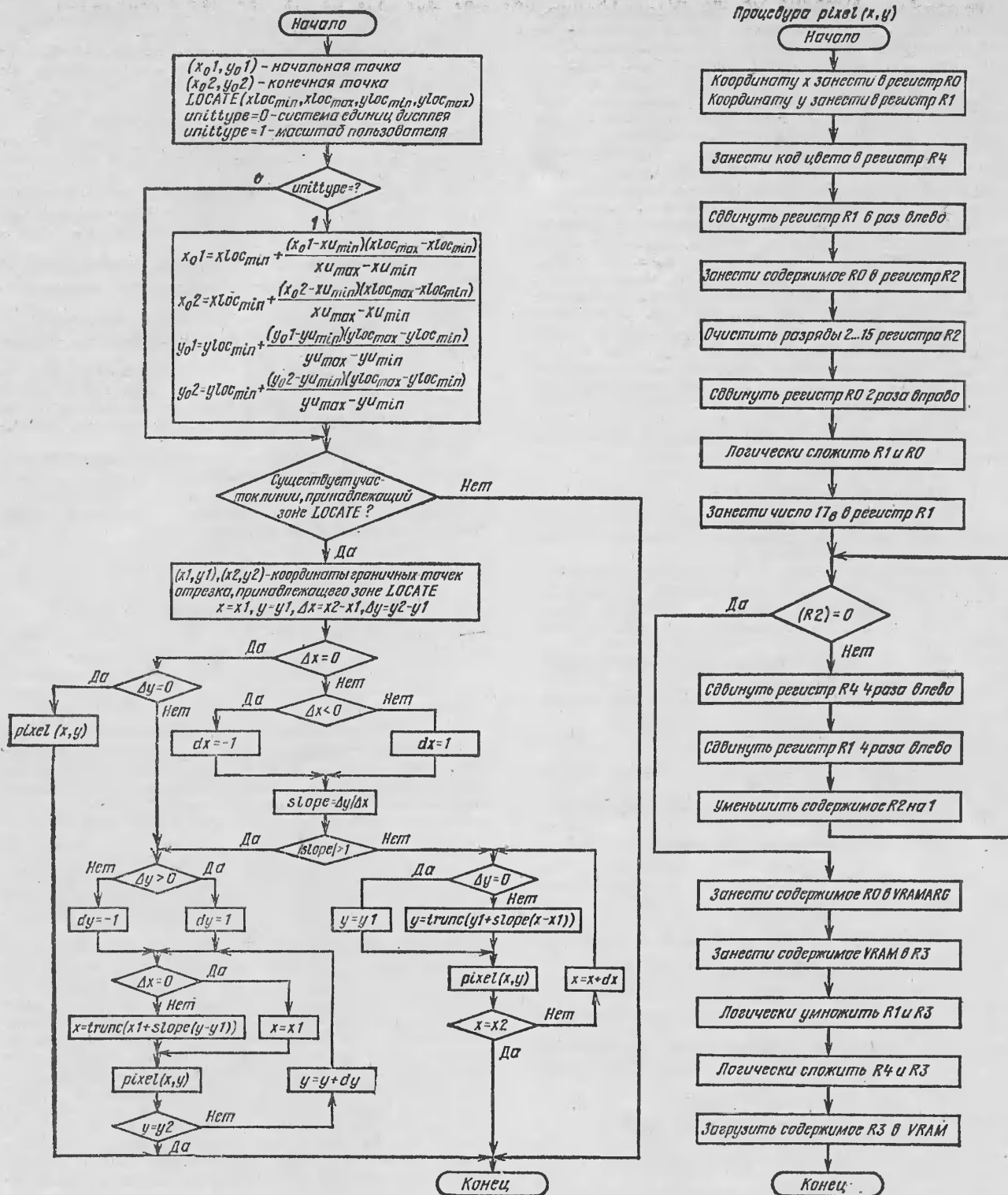


Рис. 8. Алгоритм построения отрезка прямой линии

```

procedure AXES(X0,Y0,dX,dY:real);var x0i,y0i,dxi,dyi,xt,yt,xt0,yt0,k:integer;
begin if unit_type=user_units then begin
  x0:=x_loc_min+(x0-x_user_min)*(x_loc_max-x_loc_min)/(x_user_max-x_user_min);
  y0:=y_loc_min+(y0-y_user_min)*(y_loc_max-y_loc_min)/(y_user_max-y_user_min);
  d:=dx*(x_loc_max-x_loc_min)/(x_user_max-x_user_min);
  dy:=dy*(y_loc_max-y_loc_min)/(y_user_max-y_user_min);setgu;end;
x0i:=trunc(abs(x0));y0:=trunc(abs(y0));dx:=abs(dx);dy:=abs(dy);
xt:=2;yt:=2;xt0:=xt;yt0:=yt;if x0i=x_loc_min then begin xt:=-xt;xt0:=0;end;
if x0i=x_loc_max then xt0:=0;if y0i=y_loc_min then begin yt:=-yt;yt0:=0;end;
if y0i=y_lmjj then yt0:=0;
if (x0i>x_loc_min) and (x0i<=x_loc_max) then
begin draw(x0i,ylnjj,x0i,ylmjj);y0:=abs(y0);
if y0i<y_loc_min then y0:=y_loc_min;if y0i>y_loc_max then y0:=y_loc_max;
k:=0;while (y0+k*dy)<=y_loc_max do
begin dyi:=trunc(y0+k*dy);draw(x0i+xt0,dyi,x0i-xt,dyi);k:=k+1;end;
k:=0;while (y0-k*dy)>=y_loc_min do
begin dyi:=trunc(y0-k*dy);draw(x0i+xt0,dyi,x0i-xt,dyi);k:=k+1;end;end;
if (y0i=y_loc_min) and (y0i<=y_loc_max) then
begin draw(x_loc_min,y0i,x_loc_max,y0i);x0:=abs(x0);
if x0i<x_loc_min then x0:=x_loc_min;if x0i>x_loc_max then x0:=x_loc_max;
k:=0;while (x0+k*dx)<=x_loc_max do
begin dxi:=trunc(x0+k*dx);draw(dxi,y0i+yt0,dxi,y0i-yt);k:=k+1;end;
k:=0;while (x0-k*dx)>=x_loc_min do
begin dxi:=trunc(x0-k*dx);draw(dxi,y0i+yt0,dxi,y0i-yt);k:=k+1;end;end;
if unit_type=user_units then setuu;end;

```

Рис. 9. Процедура построения осей координат

```

program SIN(X)/X_function;
var hx,x:real;i:integer;
begin graphics; clrscr;alphaoff;locate(30,460,20,350);
move(50,360);csize(3);cslant;grtext;write('Sin(x)/x function .');
scale(-20,20,-1,1);axes(0,0,1,0.1);hx:=0.05;
csize(2);cslantoff;move(-2.75,-1.05);write('-1');move(-2.0,0.95);write('1');
for i:=-2 to 2 do begin
  if i=0 then begin move(-1.5,-0.20);write(i*10:3);end
  else begin move(i*10-2.5,-0.2);write(i*10:3);end;end;
x:=-20;move(x,sin(x)/x);alphatext;
while x<=20 do begin
  if x=0 then plot(0,1) else plot(x,sin(x)/x);x:=x+hx;end;
end.

```

a

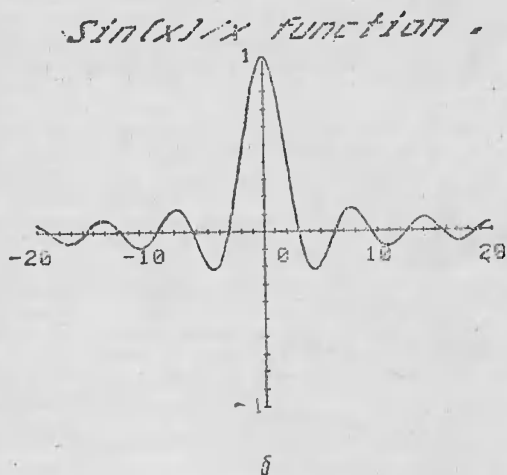


Рис. 10. Программа построения графика функции $(\sin x)/x$ (a) и график функции $(\sin x)/x$ (б)

Видеопамять загружается только на том интервале оси X, который вместе со значениями ординат прямой принадлежит зоне LOCATE. Для повышения быстродействия GRLIB часть алгоритма (рис. 8), строящая прямую на выделенном информативном интервале, написана на языке ассемблер, а остальная часть — на языке Паскаль.

Графическая библиотека GRLIB содержит широкий набор команд построения отрезков прямых линий PLOT, спиральных сегментов SPIRAL, дуг окружностей CIRCLE и интерполирующих кривых первого порядка и «n+1»-го порядка LINE и CURVE. Для первых трех операторов возможна абсолютная адресация к точке с координатами (X, Y) и адресация относительно текущего положения графического курсора (команды RPLLOT, RSPIRAL, RCIRCLE). Графический курсор без загрузки видеопамати позиционируется операторами MOVE и RMOVE. Масштабы вычисления радиусов процедур SPIRAL, PPSPIRAL, CIRCLE и RCIRCLE устанавливаются командами GRADIUS, XRADIUS, YRADIUS.

Для наглядности представления графической информации (в системах цифровой обработки сигналов и т. д.) в состав GRLIB включены операторы построения

осей координат AXES и нанесения координатной сетки GRID. Вариант упрощенной процедуры AXES, написанной на языке Паскаль и не реализующей построение больших координатных штрихов с шагом Nx/Ny, представлен на рис. 9.

Текстовая информация отображается на экране графического дисплея командами третьей группы программного знакогенератора. Параметрами знакогенератора управляют операторы TEXTDIR, CSIZE, CSLANT, CSLANTDF.

Применение библиотеки GRLIB иллюстрируется программой (рис. 10, а) построения графика функции $\sin(x)/x$ (рис. 10, б).

Рассмотренные контроллеры можно применять в составе микроЭВМ на базе любых других микропроцессоров, модифицируя системный интерфейсный блок SYSIF. Телефон для справок — 465-33-62, Москва.

УДК 681.325.5+681.326

МИКРОПРОЦЕССОРНАЯ ГРАФИЧЕСКАЯ СТАНЦИЯ ГТ-80*

В Институте технической кибернетики АН БССР создана микропроцессорная графическая станция (МГС) ГТ-80 для локально распределенной обработки данных с помощью параллельно работающих универсальной микроЭВМ, высокопроизводительного контроллера на основе секционированных микропроцессоров и аппаратных модулей.

Поскольку в составе графической станции используются векторный дисплей высокого разрешения и широкий набор периферийных устройств (графопостроитель, планшет ввода, печатающее устройство), то мощности одной универсальной микроЭВМ для управления этими устройствами, локальной модификации изображения (геометрическая трансформация, редактирование фрагментов), выполнения прикладных программ недостаточно. Для увеличения производительности станции процессы обработки децентрализуются. Выполнение каждой из выделенных процедур закрепляется за отдельным микропроцессорным элементом или аппаратным модулем, реализованным на жесткой логике (МП с фиксированной системой команд и секционированные с открытым микропрограммным уровнем). Наиболее эффективно параллельное выполнение процедур.

Универсальная микроЭВМ (связана с центральной ЭВМ) распределяет память и синхронизирует параллельные процессы, обрабатывает алфавитно-цифровую и графическую информацию, управляет работой периферийных устройств.

Микропроцессорный контроллер ведет обработку графических данных — отсечение в области просмотра на экране, установку атрибутов, управление дисплейным файлом, преобразование данных для функциональных генераторов.

Функциональные генераторы, реализованные на жесткой логике, обеспечивают вычерчивание векторов и символов на экране ЭЛТ.

Базовая модификация МГС ГТ-80 состоит из вычислительного комплекса и набора графических периферийных устройств.

Вычислительный комплекс базируется на микроЭВМ «Электроника МС 1211» [1], к которой подключаются стандартные периферийные устройства: ал-

1. Bruce A. Artwick Applied Concepts in Microcomputer Graphics. Prentice-Hall, Inc., New Jersey.—1984.—374 p.
2. Торгов Ю. И. Программируемый таймер КР580ВИ53 и его применение // Микропроцессорные средства и системы.—1984.— № 1.— С. 77—84.
3. Семенов П. А. Микроконтроллеры на базе БИС КР580 для микроЭВМ «Электроника 60» и «Электроника НЦ-80-01Д» // Микропроцессорные средства и системы.—1985.— № 3.— С. 42—45.
4. Graphics ROM For HP9848. Hewlett-Packard Desktop Computer Division, 1981.

Статья поступила 20 сентября 1986 г.

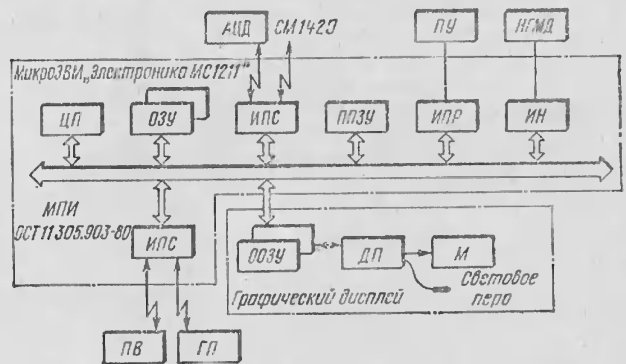
фавитно-цифровой дисплей (АЦД) 15-ИЭ-00-013, НГМД «Электроника ГМД 7012» и малогабаритное печатающее устройство (ПУ) «Электроника УВВПЧ 30-004».

Набор графических периферийных устройств включает в себя графопостроитель (ГП) ЭМ 7042 АМ со встроенным контроллером на базе одноплатной микроЭВМ, планшет ввода (ПВ) ЭМ 7029 и графический дисплей векторного типа со световым пером.

МикроЭВМ «Электроника МС 1211» — это набор модулей, объединенных системным магистральным параллельным интерфейсом (МПИ), отвечающим требованиям ОСТ 11305.903-80. В состав микроЭВМ входят: модуль 16-разрядного центрального процессора (ЦП) «Электроника МС 1601»; два модуля (ОЗУ) «Электроника МС 3101» (128К байт); модуль устройства аппаратной загрузки — диагностики (ППЗУ) «Электроника МС 3401» (до 48К байт); модуль интерфейса (ИПР) «И7» для подключения ПУ; модуль интерфейса накопителя на гибких магнитных дисках (ИН) «И4»; два модуля интерфейса последовательного (ИПС) «Электроника МС 4601» с двумя каналами в каждом для подключения АЦД, ПВ и ГП, а также для организации связи МГС ГТ-80 с центральной ЭВМ (150...19200 Бод).

Графический дисплей (см. рисунок) подключается к системной магистрали микроЭВМ. В его состав входят дисплейный процессор, общее ОЗУ (ООЗУ), цветной монитор со световым пером.

Основу дисплейного процессора (ДП) составляет унифицированный периферийный 16-разрядный контроллер ПРОМИК [2], реализованный на отечественном комплекте секционированных МП серии К1804. Тактовая частота работы контроллера 5 МГц.



Структурная схема МГС ГТ-80

* О. И. Семенов, В. В. Анищенко, А. А. Горобяченко, Е. М. Злотник, С. И. Киркоров, О. Ф. Винокурова, Н. Е. Фатеева, Е. И. Рутковский

Операционная схема включает в себя четыре каскадно связанные процессорные секции К1804ВС1, элемент ускоренного переноса К1804ВР1, одnorазрядный и байтовый двигатели, селектор 16 логических условий (флагов). В дополнение к 16 регистрам общего назначения процессорных секций в контроллере используется сверхоперативное запоминающее устройство (СОЗУ) емкостью 2К байт, время цикла 200 нс. СОЗУ временно хранит результаты вычислений, текущие атрибуты графических примитивов и системных параметров, оперативные параметры для функциональных генераторов, а также принимаемые и передаваемые в ЦП данные.

Схема управления состоит из БИС БМУ К1804ВУ4, 64-разрядного регистра микрокоманды, 8-разрядного регистра кода операции и ППЗУ для преобразования кода операции в начальный адрес микропрограммы.

Особенность контроллера — использование загружаемой оперативной памяти микропрограмм (2К 64-разрядных слов). При инициализации МПС микропрограммы, хранящиеся на гибком магнитном диске, записываются в эту память. В ЦП также имеется аппаратный модуль для оперативной отладки микропрограмм (чтения-записи оперативной памяти микропрограмм, шагового режима работы контроллера, останова по заданному адресу, запоминания в память трассы и др.). Отладка микропрограммного обеспечения поддерживается программным обеспечением САПР МПУ «МЕТАМИКРО» [3]. Благодаря этому микропрограммное обеспечение оперативно расширяется и модифицируется.

Внутренняя магистраль представляет собой высокоскоростной синхронный параллельный интерфейс, в состав которого входит 16-разрядная шина данных, две 6-разрядные шины адреса, выбирающие источник и приемник информации, шина управления. К внутренней магистрали контроллера подключены микропрограммно-управляемые аппаратные модули функциональных генераторов.

Генератор векторов — быстродействующий цифровой интерполятор, реализующий известный алгоритм аппроксимации прямых [4]. Оригинальная схема интерполятора (на интегральных элементах серии К531) обеспечивает вычисление единичного приращения за один такт (70 нс). Цифровой метод формирования векторов дает высокую точность и стабильность изображения, равномерную яркость векторов различной длины, упрощает настройку и исключает регулировку в процессе эксплуатации.

Генератор символов реализует матричный метод формирования. Основное поле знака — 7×11 точек. Точечную матрицу 158 различных символов (строчных, прописных символов русского и латинского алфавитов) хранит БИС ПЗУ К555РЕ4 (2К байт). Время развертки любого символа — постоянная величина (10 мкс). Генератор обеспечивает два размера символов, две ориентации (вертикальную и горизонтальную) и курсив (наклон на 75°) символов. Маркеры вычерчиваются в виде центрированных символов.

Восемь уровней яркости, четыре типа линий, мерцание, четыре цвета, защита фрагмента изображения от действия светового пера реализует аппаратный модуль управления атрибутами. На этом модуле имеется узел управления режимом регенерации для повторения изображения на экране ЭЛТ с фиксированными частотами 50 и 33 Гц или с частотой, определяемой размером дисплейного файла. Запрос от светового пера фиксируется специальной схемой и после вычерчивания примитива обрабатывается контроллером.

Станция ГТ-80 использует общее ОЗУ (ООЗУ). Этим ГТ-80 отличается от большинства графических дисплеев, в которых дисплейный файл хранится в ОЗУ микроЭВМ и периодически в режиме прямого доступа считывается и обрабатывается дисплейным процессором. В ООЗУ

хранятся структурированный дисплейный файл (СДФ), таблицы, специальные программы и данные [5]. ООЗУ имеет два независимых порта для подключения ЦП и ЦП. Параллельно с построением и модификацией СДФ, формированием таблиц, обработкой данных, выполняемых ЦП периодически считываются и обрабатываются в ЦП команды и данные СДФ, а на экране ЭЛТ строится графическое изображение. В ООЗУ используется схема циклического способа захвата с приоритетом для порта ЦП (среднее время ожидания — один цикл ООЗУ). ООЗУ (128К байт) реализовано на основе МПД БИС ЗУ К565РУЗ, время доступа не более 1 мкс.

Аппаратный адаптер, подключенный к системной магистрали микроЭВМ, сопрягает ЦП с ООЗУ. Команды обращения к ООЗУ — обычные процессорные. Для обращений к ООЗУ отведена зона адресов в 8К байт — так называемое окно. Содержимое 5-разрядного регистра страниц в адаптере изменяется программно при обращении к каждой новой странице ООЗУ. Такой метод адресации увеличил оперативную память микроЭВМ до 256К байт.

Дисплейный процессор с ООЗУ сопрягается через аппаратный адаптер, обеспечивающий формирование 18-разрядного адреса и управляющих сигналов.

Предвыборка данных из памяти (в адаптере), их параллельная обработка (в контроллере и функциональных генераторах) реализуют конвейер. Это значительно повышает информационную емкость изображения.

Команды дисплейного процессора — «прямые», т. е. параметры размещаются непосредственно в команде. В простейшем случае команда состоит из одного 16-разрядного слова, определяющего тип команды. Полная команда для более сложных построений и вычислительных действий состоит из типа команды и следующих за ним 16-разрядных слов, задающих параметры.

Графические команды (S-команды) формируют на экране графические примитивы с текущими атрибутами (полилиния, полимаркер, текст, короткий относительный вектор). Координаты в графических командах задаются абсолютными значениями от -16384 до $+16384$ единиц экрана и при отображении отсекаются в области просмотра на экране.

Команды установки атрибутов (S-команды) предварительно задают атрибуты графических примитивов: установить тип, цвет, ширину линий; тип, цвет, размер маркеров; тип, цвет, размер, ориентацию текста; область просмотра на экране; идентификатор выбора элемента.

Команды управления (C-команды) управляют обработкой СДФ: инициализация, очистить экран, создать фрагмент изображения с именем N, установить атрибуты фрагмента, передать управление в СДФ и др.

Команды опроса состояния (IQ-команды) передают атрибуты графических примитивов и системных параметров в ЦП.

Команды дисплейного процессора могут поступать в ЦП не только из ООЗУ, но и через адресуемые регистры ввода-вывода на системной магистрали микроЭВМ. Для обмена выделено четыре 16-разрядных регистра: регистр команд и состояния, регистр вывода, регистр ввода, регистр ошибок. Для прерывания программы в ЦП выделены 4 векторных адреса. Они обеспечивают индикацию прерывания по состоянию ЦП, обнаружению элемента (световым пером), сегмента с заданным именем, ошибки.

В цветном мониторе применена цветная ЭЛТ (пентрон) типа 61JM4Ц с электромагнитным отклонением луча и фокусировкой. Внутренняя поверхность экрана ЭЛТ покрыта люминесцентным покрытием. Цвет его свечения может изменяться от красного до зеленого и зависит от энергии электронов.

Для такой ЭЛТ разработан высоковольтный источник, формирующий в процессе вычерчивания изображения напряжения 9, 11, 13 и 15 кВ. Источник реализован на

высоковольтных транзисторах (это позволяет коммутировать цвета за 100 мкс). Узлы динамической коррекции (в мониторе) выравнивают яркость и масштаб различных цветов.

МГС ГТ-80 предназначена для работы в автономном режиме и в режиме интеллектуального графического терминала центральной ЭВМ.

Программное обеспечение (ПО) МГС ГТ-80 состоит из системного ПО, средств тестового, базового графического и прикладного ПО. Системное ПО включает ОС РАФОС с трансляторами для макроассемблера, Фортрана IV и Паскаля. Базовое графическое ПО реализует ядро графической системы (уровень 0 для языков программирования, поддерживаемых операционной системой). Ядро графической системы управляет двумя рабочими станциями — графическим дисплеем с планшетом ввода (локатором) и графопостроителем.

Тестовое ПО для МГС ГТ-80 автоматически при включении системы позволяет обнаружить неисправность в модулях микроЭВМ и графического дисплея.

В режиме терминала МГС ГТ-80 в качестве стандартного алфавитно-цифрового терминала или интеллектуального графического эмулирует системный терминал ЭВМ СМ1420. Для этого в оперативную память микроЭВМ с ГМД загружается программа приема и обработки информации от центральной ЭВМ, управления планшетом ввода, графопостроителем и графическим дисплеем. В среде ОС РВ СМ1420 имеется библиотека процедур ядра графической системы для языков Фортран IV и Паскаль. Входной язык МГС ГТ-80 — уровень интерфейса виртуального устройства. На ЭВМ СМ1420 исполняются прикладные пакеты, использующие интерфейс ядра графической системы.

Характеристики графического дисплея МГС ГТ-80 приведены в сравнении с характеристиками графического дисплея ГРАФИТ. Его управляющая микроЭВМ — «Электроника 60», программно и аппаратно совместима с микроЭВМ «Электроника МС 1211».

Технические характеристики графопостроителя ЭМ 7042 АМ

Размер рабочего поля, мм	420×300
Скорость вычерчивания, мм/с	400
Число смещных инструментов	4

УДК 681.3.06

А. Н. Алешин, С. Н. Крюков

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИМВОЛЬНО-ГРАФИЧЕСКОГО ЦВЕТНОГО ТЕЛЕВИЗИОННОГО ДИСПЛЕЯ

Функциональные модули «Электроника С5-2106» [1] ИУЗ.057.244 и «Электроника С5-2106А» ИУЗ.057.244-01 легко подключаются к микроЭВМ «Электроника 60», так как в них используется интерфейс МПИ по ОСТ 11.305.903-80. Путем несложной доработки любой цветной бытовой телевизор достаточно просто сопрягается с этими модулями. Таким образом, можно получить гибкую растровую систему отображения цветной видеографической информации.

Технические данные такой системы позволяют вывести на экран телевизора символьную и графическую информацию в семи цветах (синем, зеленом, красном, голубом, желтом, пурпурном и белом) с разрешением 256 строк по 512 элементов в каждой.

Технические характеристики планшета ввода ЭМ 7029

Размер рабочего поля, мм	420×300
Разрешающая способность, мм	0,25
Погрешность ввода координат, мм	±0,1

Технические характеристики графических дисплеев

	МГС ГТ-80	ГРАФИТ
Размер экрана, мм	310×310	340×340
Число цветов	4	1
Число типов линий	7	7
Информационная емкость изображения (объем информации, выводимый без мерцания):		
векторов длиной до 32 мм, шт.	Не менее 2300	Не менее 1100
векторов длиной свыше 256 мм, шт.	Не менее 275	Не менее 250

Конструктивно ДП и ООЗУ графического дисплея МГС ГТ-80 размещаются в одном субблоке, в состав которого входят 14 модулей на платах (150×320 мм): модули контроллера — 5 плат, ООЗУ — 5, функциональных генераторов — 4.

Адрес для справок: 220605, г. Минск, ул. Сурганова, 6, ИТК АН БССР. Тел. 39-59-85.

ЛИТЕРАТУРА

1. Лопатин В. С., Пархоменко П. И., Токмаков В. И. МикроЭВМ «Электроника-МС 1211», «Электроника МС 1212» // Микропроцессорные средства и системы. — 1985. — № 2. — С. 14—15.
2. Злотник Е. М. Унифицированный периферийный контроллер ПРОМИК на базе секционированных микропроцессоров. — Тр. конф. «Микросистема-85». — ЧССР, Табор, 1985, с. 314—317.
3. Семенов О. И. и др. Настраиваемый инструментальный комплекс для разработки систем на секционированных микропроцессорах // УСИМ. — 1984. — № 2. — С. 36—39.
4. Фоли Дж., ван Дэм А. Основы интерактивной машинной графики. Т. 2. — М.: Мир, 1985. — С. 140—143.
5. Киркоров С. И. Организация дисплейного файла. — В сб.: Теория и методы автоматизации проектирования. — Минск, 1984. — Вып. 4. — С. 68—73.

Статья поступила 2 февраля 1987 г.

Функциональная схема системы и схема сопряжения представлены на рис. 1 и 2. Номера разъемов на рисунке соответствуют принятым в [1]. Подходит любой телевизор с кинескопами типа 59 ЛК3Ц или 61 ЛК3Ц. Устройство сопряжения представляет собой три видеусилителя сигнала цветности В_г, В_с, В_в, вырабатываемых модулем, и подключается по выходу непосредственно к соответствующим катодам кинескопа телевизора. Потенциометрами R₂ регулируется баланс белого, R₆ — уровень черного и R₃ — яркость изображения. Пи-

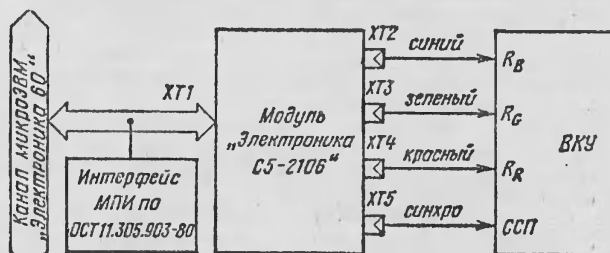


Рис. 1. Функциональная схема связи модуля с микроЭВМ и цветным видеоконтрольным устройством (ВКУ)

тающие схему напряжения можно получить от внешних источников или от блока питания используемого телевизора.

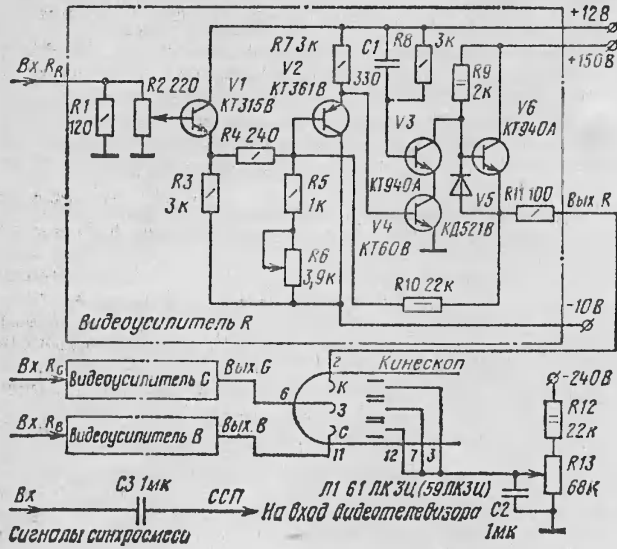


Рис. 2. Схема сопряжения модуля с телевизором

Сигналы синхросмеси, формирующие растр изображения, подаются через емкость C_3 на вход видеотелевизора. Блок цветности телевизора не используется и может быть изъят из аппарата.

Разработан пакет программ, написанный на языке ассемблера микроЭВМ «Электроника 60», обеспечивающий программирование и управление дисплейным модулем «Электроника С5-2106» в символьном и графическом режимах.

Пакет реализует функции:

- SI □ — программирование регистров модуля в символьном режиме;
- GR □ — программирование регистров модуля в графическом режиме;
- CLRMEM — очистка видеопамати модуля;
- SYMBOL — отображение символа, заданного размера и цвета в заданное место экрана, с возможностью включения мерцания символа, инверсии подсчета знакоместа, подчеркивания;
- POINT — отображение точки заданным цветом с геометрическими координатами X и Y с возможностью включения мерцания;
- GLINE — подсвет заданным цветом горизонтальной линии с координатой Y;
- ULINE — подсвет заданным цветом вертикальной линии с координатой X;
- CURVE — подсвет заданным цветом кривой, заданной в виде массива {X, Y};
- TEXT — отображение в требуемом месте на экране текста в заданном формате и цвете;
- CVADRO — подсвет квадрата с заданными границами;
- STOLB — подсвет столбика с заданными границами.

К разработанному функциональному пакету пользователь обращается из прикладной программы, написанной на языке QUASIC [2], хотя тип языка, на котором идет решение основной программы, не существен, важно лишь, чтобы этот язык имел бы возможность обращаться к подпрограммам, написанным на ассемблере и обмениваться с ними параметрами.

Форма обращения к пакету из системы QUASIC:
CALLI, FUN, A1, A2, ..., AN, ERR,

```

00001 ;
00002 ; ***** АССЕМБЛЕР ЭЛЕКТРОНИКА-60М / ПРОГРАММИСТ-2 *****
00003 ;
00004 ; POINT - ЗАПИСЬ ИНФОРМАЦИИ В ПАМЯТЬ АДАПТЕРА *
00005 ;
00006 ;
00007 ; ВХОДНЫЕ ПАРАМЕТРЫ
00008 ; X - КООРДИНАТА "X" ТОЧКИ ГРАФИКА
00009 ; Y - КООРДИНАТА "Y" ТОЧКИ ГРАФИКА
00010 ; ЦВЕТ - ЦВЕТ ТОЧКИ
00011 ; 0 - ГАШЕНИЕ
00012 ; 1 - СИНИЙ
00013 ; 2 - ЗЕЛЕНЬ
00014 ; 3 - ГОЛУБЫЙ
00015 ; 4 - КРАСНЫЙ
00016 ; 5 - ПУРПУРНЫЙ
00017 ; 6 - ЖЕЛТЫЙ
00018 ; 7 - БЕЛЫЙ
00019 ;
00020 ; MER - ВКЛЮЧЕНИЕ МЕРЦАНИЯ
00021 ; 1 - ДА
00022 ; 0 - НЕТ
00023 ;
00024 ;
00025 ; ZONE - ЗОНА ПАМЯТИ НА ЗАПИСЬ
00026 ; 0,4 - НОМЕР ЗОНЫ
00027 ;
00028 ;
00029 ;
00030 ; ***** ЧТЕНИЕ ВХОДНЫХ ПАРАМЕТРОВ *****
00031 ;
00032 POINT: MOV     0(R5)+,XP      ;КООРДИНАТА "X"
00033 MOV     0(R5)+,YP      ;КООРДИНАТА "Y"
00034 MOV     0(R5)+,CVET    ;ЦВЕТ
00035 MOV     0(R5)+,MER     ;МЕРЦАНИЕ
00036 MOV     0(R5)+,ZONE    ;ЗОНА ПАМЯТИ
00037 ;
00038 ; ***** ПРОВЕРКА ЗНАЧЕНИЙ ПЕРЕМЕННЫХ *****
00039 ;
00040 POINT1: TST     XP      ;ПРОВЕРКА
00041 BPL     P06           ; КООРДИНАТЫ "X"
00042 P05:  MOV     #2,ERR   ;
00043 JMP     END           ;
00044 P06:  CMP     XMAX,XP  ;
00045 BPL     P07           ;
00046 BR     P05           ;
00047 P07:  TST     YP      ;ПРОВЕРКА
00048 BPL     P09           ; КООРДИНАТЫ "Y"
00049 P08:  MOV     #3,ERR   ;
00050 JMP     END           ;
00051 P09:  CMP     YMAX,YP  ;
00052 BPL     P0100        ;
00053 BR     P08           ;
00054 P0100: TST     ZONE   ;ПРОВЕРКА
00055 BEQ     POINT2      ; НОМЕРА ЗОНЫ
00056 CMP     #4,ZONE     ; ПАМЯТИ
00057 BEQ     POINT2      ;
00058 MOV     #4,ERR      ;
00059 JMP     END          ;
00060 ;
00061 ; ***** ОПРЕДЕЛЕНИЕ АДРЕСА ЯЧЕЙКИ ПАМЯТИ *****
00062 ;
00063 POINT2: MOV     XP,R0      ;R0=XP/8
00064 MOV     #3,R1          ;
00065 P011:  ASR     R0        ;
00066 SOB     R1,P011       ;
00067 MOV     YMAX,R2       ;R2=YMAX-YP
00068 SUB     YP,R2         ;
00069 CMP     XMAX,#511     ;R2=(YMAX-YP)*(XMAX+1)/8
00070 BEQ     P012         ;
00071 MOV     #5,R1        ;
00072 BR     P013         ;
00073 P012: MOV     #6,R1   ;
00074 P013: ASL     R2       ;
00075 SOB     R1,P013     ;
00076 ADD     R0,R2       ;
00077 MOV     R2,AL        ;AL=X/8+(YMAX-YP)*(XMAX+1)
00078 ;
00079 ; ***** ОПРЕДЕЛЕНИЕ ЗОНЫ ПАМЯТИ И *****
00080 ; ***** АДРЕСА ЯЧЕЙКИ В ЗОНЕ *****
00081 ;
00082 CLR     R0A           ;
00083 MOV     #2047,R0     ;
00084 P014: CMP     R0,AL  ;
00085 BPL     P015       ;

```

```

00086 ZNC RGA ;В RGA-ОТНОС. НОМЕР ЗОНЫ
00087 SUB #2048,AL ;В AL-АДРЕС ЯЧЕЙКИ В ЗОНЕ
00088 BR P014 ;
00089 P015: ADD ZONE,RGA ;
00090 ;
00091 ; ***** ОПРЕДЕЛЕНИЕ НОМЕРА ТОЧКИ В ПИКСЭЛЕ *****
00092 ;
00093 MOV XР,R0 ;В R0-НОМЕР ТОЧКИ В ЯЧЕЙКЕ
00094 BIC #17770,R0 ;
00095 MOV #8.,R2 ;R2=8.-R0
00096 SUB R0,R2 ;
00097 CLR R1 ;
00098 ; ;В R1-НОМЕР РАЗРЯДА
00099 P016: ROL R1 ;
00100 SOB R2,P016 ;

```

TV.C5 21-06 STR 00004

```

00101 ;
00102 ; ***** ЧТЕНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ ПАМЯТИ *****
00103 ;
00104 CMP AL,RENAL ;ОБРАЩЕНИЕ В ТУ ЖЕ ЯЧЕЙКУ ?
00105 BNE P017 ;НЕТ
00106 BR P018 ;ДА
00107 P017: CLR RMEM ;
00108 ;
00109 ; ***** ЗАДАНИЕ СОДЕРЖИМОГО ЯЧЕЙКИ *****
00110 ;
00111 P018: CMP R1,#17 ;
00112 BGT P024 ;1-й ПИКСЭЛ
00113 ;
00114 BIT #400,RMEM ;ЕСТЬ МЕРЦАНИЕ ?
00115 BNE P023 ;УЖЕ ЕСТЬ
00116 TST CVET ;ПОДСВЕТ ИЛИ ГАШЕНИЕ ?
00117 BNE P020 ;ПОДСВЕТ
00118 ;
00119 P019: BIC R1,RMEM ;ГАШЕНИЕ ТОЧКИ
00120 BR P028 ;
00121 ;
00122 P020: MOV CVET,UPR ;ФОРМИРОВАНИЕ
00123 ASL UPR ;УПРАВЛЕНИЯ
00124 ADD MER,UPR ; 2-М ПИКСЭЛОМ
00125 MOV #8.,R2 ;
00126 P021: ASL R2,P021 ;
00127 SOB R2,P021 ;
00128 BIC #7400,RMEM ;
00129 BIS UPR,RMEM ;
00130 ;
00131 P022: BIS R1,RMEM ;ПОДСВЕТ ТОЧКИ
00132 BR P028 ;
00133 ;
00134 P023: TST CVET ;ПОДСВЕТ ИЛИ ГАШЕНИЕ ?
00135 BNE P022 ;ПОДСВЕТ
00136 BR P019 ;ГАШЕНИЕ
00137 P024: BIT #10000,RMEM ;ПРОВЕРКА НАЛИЧИЯ МЕРЦАНИЯ
00138 BNE P027 ;УЖЕ ЕСТЬ
00139 TST CVET ;ПОДСВЕТ ИЛИ ГАШЕНИЕ ?
00140 BNE P025 ;ПОДСВЕТ
00141 BR P019 ;ГАШЕНИЕ
00142 P025: MOV CVET,UPR ;ФОРМИРОВАНИЕ
00143 ASL UPR ;УПРАВЛЕНИЯ
00144 ADD MER,UPR ; 1-М ПИКСЭЛОМ
00145 MOV #12.,R2 ;
00146 P026: ASL UPR ;
00147 SOB R2,P026 ;
00148 BIC #170000,RMEM ;
00149 BIS UPR,RMEM ;
00150 BR P022 ;

```

TV.C5 21-06 STR 00004

```

00151 P027: TST CVET ;ПОДСВЕТ ИЛИ ГАШЕНИЕ ?
00152 BNE P022 ;ПОДСВЕТ
00153 BR P019 ;ГАШЕНИЕ
00154 ;
00155 ; ***** ЗАПИСЬ В ЯЧЕЙКУ ПАМЯТИ *****
00156 ;
00157 P028: CMP AL,RENAL ;ОБРАЩЕНИЕ В ТУ ЖЕ ЯЧЕЙКУ ?
00158 BEQ P029 ;ДА
00159 MOV AL,RENAL ;НЕТ
00160 P029: ASL AL ;
00161 MOV @BEGMEM,ADRMEM ;
00162 ADD AL,ADRMEM ;
00163 BIC #7,A2 ;
00164 BIS RGA,A2 ;
00165 CLR NINT ;
00166 MOVB A2,@RGA2 ;
00167 CLR NINT ;
00168 MOV RMEM,ADRMEM ;
00169 JMP END ;

```

где I — номер подключаемой ассемблерной подпрограммы (в данном случае пакета); FUN — функция, реализуемая пакетом; A1, A2, ..., AN — параметры функции; ERR — результат обращения к функции (ERR=0 — функция выполнена, ERR=1 — модуль не откликается, ERR=i — ошибки в задаваемых параметрах, значение i определяется конкретной функцией).

Функция распознана по первым двум символам названия функции. При обращении к несуществующей функции происходит аварийный останов с печатью диагностического текста: «Пакет TV.C5 — несуществующая функция...»

Фрагмент программы, организующий распознавание выполняемой функции и ее инициализацию (см. рис. 3).

Рис. 4

```

00001 ; АССЕМБЛЕР ЭЛЕКТРОНИКА-60М / ПРОГРАММИСТ-2
00002 ; *****
00003 ; TV.C5 - ПАКЕТ ПОДПРОГРАММ, ПОЗВОЛЯЮЩИХ ПРОГ-
00004 ; РАММИРОВАТЬ ЦВЕТНОЙ ДИСПЛЕЙНЫЙ АДАПТЕР C5 21-06 В
00005 ; СИМВОЛЬНОМ И ГРАФИЧЕСКОМ РЕЖИМАХ.
00006 ;
00007 ; ФУНКЦИИ, РЕАЛИЗУЕМЫЕ ПАКЕТОМ:
00008 ; SIA - ПРОГРАММИРОВАНИЕ РЕГИСТРОВ АДАПТЕРА В СИМ-
00009 ; ВОЛЬНОМ РЕЖИМЕ
00010 ; ORX - ПРОГРАММИРОВАНИЕ РЕГИСТРОВ АДАПТЕРА В ГРА-
00011 ; ФИЧЕСКОМ РЕЖИМЕ
00012 ; CLRMEM - ОЧИСТКА ПАМЯТИ АДАПТЕРА
00013 ; SYMBOL - ЗАПИСЬ СИМВОЛА
00014 ; POINT - ПОДСВЕТ ТОЧКИ
00015 ; GLINE - ПОДСВЕТ ГОРИЗОНТАЛЬНОЙ ЛИНИИ
00016 ; VLINE - ПОДСВЕТ ВЕРТИКАЛЬНОЙ ЛИНИИ
00017 ; CURVE - ПОДСВЕТ КРИВОЙ
00018 ; TEXT - ЗАПИСЬ ТЕКСТА
00019 ; CVADRO - ПОДСВЕТ КВАДРАТИКА
00020 ; STOLB - ПОДСВЕТ СТОЛБИКА
00021 ;
00022 ; ФОРМА ОБРАЩЕНИЯ К ПАКЕТУ ИЗ СИСТЕМЫ "QUASIC"
00023 ; CALL N;FUN,A1,.....,AJ,ERR ;
00024 ; ГДЕ FUN-ФУНКЦИЯ, РЕАЛИЗУЕМАЯ ПАКЕТОМ
00025 ; A1,.....,AJ-ПАРАМЕТРЫ ФУНКЦИИ
00026 ; ERR-РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ФУНКЦИИ:
00027 ; ERR=0-ФУНКЦИЯ ВЫПОЛНЕНА
00028 ; ERR=1-АДАПТЕР НЕ ОТВЕЧАЕТ
00029 ; ERR=I-ОШИБКИ В ПАРАМЕТРАХ, ЗНАЧЕНИЕ I
00030 ; ОПРЕДЕЛЯЕТСЯ ФУНКЦИЕЙ
00031 ;
00032 ; РАСПОЗНАВАНИЕ ФУНКЦИИ ПРОИСХОДИТ ПО ПЕРВЫМ ДВУМ
00033 ; СИМВОЛАМ НАЗВАНИЯ ФУНКЦИИ. ПРИ ОБРАЩЕНИИ К НЕСУЩЕСТ-
00034 ; ВУЩЕЙ ФУНКЦИИ ПРОИСХОДИТ АВАРИЙНЫЙ ОСТАНОВ С ПЕЧАТЬЮ
00035 ; "ПАКЕТ TV.C5 - НЕСУЩЕСТВУЮЩАЯ ФУНКЦИЯ ....."
00036 ; *****
00037 ;
00038 .TITLE TV.C5
00039 .GLOBL TV.C5
00040 ;
00041 R0=X0
00042 R1=X1
00043 R2=X2
00044 R3=X3
00045 R4=X4
00046 R5=X5
00047 SP=X6
00048 PC=X7
00049 RGA1=167740 ;РЕГ. УПРАВЛЕНИЯ
00050 RGA2=167744 ;УПР.ПАМЯТЬ

```

TV.C5 21-06 STR 00002

```

00051 RSB1=177740 ;ОБР. ХОД ПО СТРОКЕ
00052 RGY1=177742 ;ПРЯМ. ХОД ПО СТРОКЕ
00053 RGB2=177744 ;СТРОКИ НАД СИМВОЛОМ
00054 RGY2=177746 ;СТРОКИ ПОД И В СИМВОЛЕ
00055 RGB3=177750 ;ОБР. ХОД ПО КАДРУ
00056 RGY3=177752 ;ПРЯМ. ХОД ПО КАДРУ
00057 RSTT=177564 ;РС ПЕЧАТИ ТЕРМИНАЛА
00058 RDTT=177566 ;РД ПЕЧАТИ ТЕРМИНАЛА
00059 INTMAX=1000. ;МАКС. ЧИСЛО ПЕРЫВАНИЙ
00060 BEGMEM=150000 ;НАЧ. АДРЕС ПАМЯТИ
00061 ;
00062 ; ***** ОПРЕДЕЛЕНИЕ ВЫПОЛНЯЕМОЙ ФУНКЦИИ *****
00063 ;
00064 TV.C5: MOV @R5+,R0 ;НАЗВАНИЕ ФУНКЦИИ
00065 MOV @TANFUN,R1 ;НАЧАЛО ТАБЛИЦЫ
00066 TV1: TST (R1) ;КОНЕЦ ТАБЛИЦЫ ?
00067 BNE TV3 ;НЕТ

```



```

00068 MOV $BUF1,R1 ;ОБЪЕКА
00069 MOV R0,#40,(R1) ; ПЕЧАТЬ
00070 MOV $A6,R0 ; ДИАГНОСТИЧЕСКОГО
00071 TV2: TSTB @RSTT ; ТЕКСТА
00072 BPL TV2 ;
00073 MOVSB (R1)+,@$RDTT ;
00074 SOB R0,TV2 ;
00075 HALT ;АВАРИЙНЫЙ ОСТАНОВ
00076 ;
00077 TV3: CMP (R1)+,R0 ;НАЗВАНИЕ НАЙДЕНО ?
00078 BNE TV1 ;НЕТ
00079 CLR ERR ;ПОДГОТОВКА
00080 MOV $INTER,@$4 ;
00081 MOV $200,@$6 ;
00082 CLR NINT ;
00083 ADD $22,R1 ; ПЕРЕХОД К ВЫПОЛНЕНИЮ
00084 MOV (R1),R1 ; ФУНКЦИИ
00085 JMP (R1) ;
00086 ;
00087 TABFUN: .ASCII /SX/ ;ТАБЛИЦА ИМЕН
00088 .ASCII /BX/ ;
00089 .ASCII /CL/ ;
00090 .ASCII /SI/ ;
00091 .ASCII /PO/ ;
00092 .ASCII /GL/ ;
00093 .ASCII /VL/ ;
00094 .ASCII /CU/ ;
00095 .ASCII /TE/ ;
00096 .ASCII /CV/ ;
00097 .ASCII /ST/ ;
00098 0 ;КОНЕЦ ТАБЛИЦ ИМЕН
00099 ADRFUN: SIK ;ТАБЛИЦА АДРЕСОВ
00100 GRX ;

```

TV.C5 21-06 СТР 00003

```

00101 CLRMEM ;
00102 SYMBOL ;
00103 POINT ;
00104 GLINE ;
00105 VLINE ;
00106 CURVE ;
00107 TEXT ;
00108 CVADRO ;
00109 STOLB ;
00110 ;
00111 BUF1: .BYTE 15,12
00112 .ASCII /"ПАКЕТ TV.C5 - НЕСУЩЕСТВУЮЩАЯ ФУНКЦИЯ
00113 .BYTE 0,0,40,42,15,12
00114 ;
00115 INTER: INC NINT ;СЧЕТЧИК ПРЕРЫВ.
00116 CMP $INTMAX,NINT ;INTMAX > NINT ?
00117 BPL INT1 ;ДА
00118 MOV $1,ERR ;
00119 CMP (SP)+,@(R5)+ ;
00120 INT2: TSG IN ;
00121 BEQ TV4 ;
00122 TST (6P)+ ;
00123 DEC IN ;
00124 BR INT2 ;
00125 INT1: SUB $6,(SP) ;АДРЕС ВОЗВРАТА
00126 RTI ;
00127 ;
00128 END: TST IN ;ВНУТРЕННИЙ ВЫЗОВ ?
00129 BEQ TV4 ;НЕТ
00130 DEC IN ;
00131 RTS PC ;
00132 TV4: MOV ERR,@(R5)+ ;
00133 RTS R5 ;
00134 ;
00135 .END ;

```

Два символа названия функции в кодах ASCII передается из вызывающей программы в регистр R0 с помощью оператора под меткой TV.C5:

Особенность данной программы — использование прерываний при отсутствии отклика модуля при обращении к его внутренним регистрам или к видеопамяти через общую зону адресов. Опыт совместной эксплуатации модуля «Электроника С5-2106 с микроЭВМ «Электроника 60М» показал, что часто при обращении к внутренним адресам модуля по техническим причинам не происходит требуемого отклика модуля и микроЭВМ переходит к прерыванию по несуществующему адресу. Вектор этого прерывания записан по адресу 4. При

многократном повторном обращении к тому же самому внутреннему адресу модуля удается осуществить связь ЭВМ с модулем. Для этого по адресу 4 заносится вектор прерывания, обеспечивающий обращение к подпрограмме обработки прерывания INTER. Подпрограмма обеспечивает многократное повторное обращение к модулю с числом попыток INTMAX. Число INTMAX для конкретного модуля определяется опытным путем. В качестве примера приведена подпрограмма POINT — запись информации в видеопамять модуля в графическом режиме его работы (см. рис. 4). Обращение к подпрограмме имеет вид

CALLI, PO, X, Y, CVET, MER, ZONE, ERR

Обозначение входных параметров понятно из приведенных в программе комментариев. Начинается выполнение подпрограммы с метки POINT приемом входных параметров.

Часть программы от метки POINT1 до метки PO9 анализирует величины XP и YP и в случае выхода их за допустимые границы присваивает переменной ERR соответствующие значения и прекращает выполнение подпрограммы POINT, осуществляя выход в конце пакета на метку END:

ERR=2 при XP>XMAX или XP<0,
ERR=3 при YP>YMAX или YP<0.

Очевидно, переменная YP должна иметь размерность числа строк, а XP — числа точек в строке. Начало координат расположено в левом нижнем углу телеэкрана. Величины XMAX и YMAX формируются заранее подпрограммой GR X программирующей регистры модуля в графическом режиме.

Начиная с метки POINT2 до метки PO15 включительно, рассчитывается абсолютный номер ячейки видеопамяти модуля AL, соответствующий заданным XP и YP:

$$AL = X/8 + (YMAX - YP) \cdot (XMAX + 1)/8.$$

При выводе формулы учитывалось, что в каждой ячейке памяти содержится информация о восьми точках.

Далее, учитывая, что видеопамять модуля состоит из восьми зон по 2К слов в каждой, номер ячейки представляется в виде PGA — номера зоны видеопамяти и AL — относительного номера ячейки в соответствующей зоне.

Структура слова для графического режима работы модуля следующая [1]:

Байт	Управляющий (УПР)							
	15p	14p	13p	12p	11p	10p	9p	8p
Назначение	УПР 1 пиксел*				УПР 2 пиксел			
	к	з	с	ме	к	з	с	ме

Продолжение

Байт	Информационный (подсвет точек (ТЧК))							
	7p	6p	5p	4p	3p	2p	1p	0p
Назначение	ТЧК 1 пиксел				ТЧК 2 пиксел			
	к	з	с	ме	к	з	с	ме

Примечание: к — красный, з — зеленый, с — синий, ме — мерцание с частотой 1...3 Гц.

* Пиксел — это минимальная группа точек, цветом и мерцанием которой можно управлять.

Начиная со второго оператора после метки PO16 и до метки PO16, в регистре R1 рассчитывается положение бита подсвета искомой точки в соответствующей ячейке памяти.

Далее определяется соответствие подсвечиваемой точки одному из двух пикселей в ячейке; в случае первого пикселя управление передается на метку PO24, где формируется слово управления — PMEM в соответствии со структурой [1] и входными параметрами X, Y, CVET, MER, после чего управление передается на метку PO29.

В этом месте программы относительный номер ячейки AL пересчитывается в относительный адрес: $AL = 2 \cdot AL$, а затем в абсолютный адрес микроЭВМ — ADRMEM, находящийся в общем с модулем адресном поле:

$$ADRMEM = BEGMEM + AL,$$

где BEGMEM — начало общей зоны памяти (адрес модуля) задается распайкой соответствующих контактов в самом модуле [1].

УДК 681.3.022

П. А. Бабкин, О. П. Солоненко, Б. В. Тарасов,
В. Г. Федорин

СРЕДСТВА ЦВЕТНОЙ МАШИНОЙ ГРАФИКИ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Современное направление [1, 2] в области автоматизации научных исследований предполагает широкое использование вычислительных комплексов на базе различных микроЭВМ, программно и аппаратно совместимых с микроЭВМ «Электроника 60». В этой связи весьма актуальны разработки мобильных программных средств для отображения цветной графической информации с использованием ТВ-мониторов [3], являющихся периферийными устройствами микроЭВМ «Электроника 60».

Настоящая подсистема цветной машинной графики (ЦМГ) ориентирована на устройство управления монитором «Электроника МС 4702», получившее широкое распространение при отображении информации на базе телевизионных приемников, применяемых как периферийные устройства микроЭВМ «Электроника 60», МС 1201 и других программно и конструктивно совместимых с ними. Подсистема ЦМГ состоит из двух пакетов программ и графического редактора. Пакеты ЦМГ написаны на языке ассемблера MACRO-11 и, как объектные библиотеки, расположенные на гибком диске пользователя, доступны для программ, разрабатываемых на языке Фортран. Подсистема ЦМГ функционирует в рамках операционной системы РАФОС, объем ядра пакетов не превышает 3,5К слов оперативной памяти при одновременном использовании всех подпрограмм одного пакета. Графический редактор [4], разработанный как Фортрановская программа на базе первого пакета ЦМГ, позволяет с помощью клавиатуры терминала, служащего системным пультом, осуществлять в диалоговом режиме редактирование или создание рисунка с последующей фиксацией его в виде дискового файла.

Устройство управления монитором (контроллер) «Электроника МС 4702» предназначено для работы в составе микроЭВМ, аналогичных широко распространенной «Электроника 60М», для управления цветным графическим монитором. В составе ЭВМ контроллер работает как асинхронное пассивное устройство, т. е. выполняет команды, генерируемые центральным процессором. Способ формирования изображения — растровый, емкость буферной памяти контроллера 192К бит (3 бит на одну точку раstra). Число точек изображения 256×256 , отображаемых цветов — 8.

По адресу регистра 2А модуля [1] передается номер зоны видеопамати, к которой происходит обращение со стороны магистрали ЭВМ. В программе этот адрес обозначен как RGA2, а по адресу ADRMEM передается информационное слово PMEM, обеспечивающее подсветку точки с координатами XР, YР.

Рассмотренный пакет программ занимает объем ОЗУ микроЭВМ 2К слов.

Телефон для справок: 516-99-29, Ленинград

ЛИТЕРАТУРА

1. Модуль функциональный «Электроника С5-2106». Техническое описание. ИУЗ.057.244.
2. Подольский Л. И. Система QUASIC для программирования на мини-ЭВМ. — Пушкино: Науч. центр биолог. исследований АН СССР. — 1980, вып. 4.

Статья поступила 30 июня 1986 г.

Имеется возможность построения вертикальных, горизонтальных, диагональных векторов посредством вращенного генератора. Точка выводится за время двух обращений ЭВМ к контроллеру, а в составе вектора — за время одного обращения ЭВМ к контроллеру.

Программные средства ЦМГ состоят из двух пакетов подпрограмм. Основа первого пакета — система программного обеспечения графопостроителей (СМОГ) [5]. Пакет состоит из 19 подпрограмм (OPENED, INFOR, LIST, PERO, DEK, ERASE, CLEALL, TRA, TEXT, CHAR, POINT, LIK, DECAR, LID, DEL, TRAD, VAL, POIN, CLOSED), составляющих его ядро, по существу, адекватное первому уровню СМОГ, и двух служебных подпрограмм, предназначенных для вывода на экран монитора рисунков, хранящихся в архиве в виде дисковых файлов. Пакет предоставляет пользователю два режима работы: непосредственного вывода информации на экран монитора и имитации, позволяющий организовать архив рисунков, имеющий структуру дисковых файлов ОС РАФОС.

Работа с пакетом иницируется OPENED. Выделение области рисования, или листа, производит LIST. Каждое повторное обращение к LIST выделяет новую область рисования, а прежняя область становится недоступной. В области рисования LIST автоматически вводит листовую систему координат, PERO устанавливает цвет, которым рисуют все изобразительные подпрограммы пакета. ERASE стирает лист цветом, установленным PERO, CLEALL стирает весь экран. POINT (X, Y) высвечивает точку экрана с листовыми координатами (X, Y) цветом, установленным PERO. TRA (X, Y, K) и TRAD (X, Y, K) проводят отрезок прямой цветом (K), установленным PERO. Началом отрезка считается местоположение «текущей» точки листовых координат.

Для изображения стандартных символов кода ASCII с единственным масштабом резервируется литерная площадка размером 6×8 точек. Масштаб выдаваемых символов может быть целым числом от 1 до 32. CHAR выводит на экран монитора один символ, закодированный в первом параметре, масштаб выдаваемого символа задается вторым параметром подпрограммы.

С помощью TEXT на экран выводятся символы, заданные в некотором массиве. VAL используется для вывода на экран монитора действительного числа, заданного первым параметром, в формате, определяемом вторым параметром подпрограммы.

Первый пакет ЦМГ используется при отображении информации в автоматизированной системе экспериментальных исследований ИТФ СО АН СССР, а также применяется для визуализации результатов вычислительных экспериментов.

Для задач, связанных с моделированием физических процессов, потребовалось разработать более быстродействующее программное средство. Исходя из требований

обеспечения максимального быстродействия и минимальной занятости оперативной памяти микроЭВМ «Электроника 60», был разработан в рамках описываемой системы ЦМГ второй пакет подпрограмм, позволяющий работать только с физическими координатами экрана монитора. Это позволило значительно увеличить скорость рисования за счет использования возможности вывода точки в составе вектора, т. е. за время одного обращения ЭВМ к контроллеру, а также за счет полного отказа от «плавающей» арифметики.

Второй пакет ЦМГ состоит из 10 подпрограмм: PEROF, CLEAF, POIF, TRAF, CHARF, TEXTF, VALF, TRALF, COLOR, SWAPP и двух специальных подпрограмм SHIM и WSHIM. Этот пакет работает с единственной системой координат — системой физических координат экрана, являющейся частным случаем листовых координат, когда область рисования является все рабочее поле монитора. PEROF устанавливает текущий активный цвет, используемый в CLEAF, CHARF, TEXTE, VALF, TRALF. CLEAF стирает экран монитора. POIF (X, Y, C) производит вывод точки, где $0 \leq X, Y \leq 255$ — ее физические координаты, CL — цвет выводимой точки.

TRAF (X, Y, DL, N, CL) производит построение вектора, где X, Y — физические координаты начальной точки вектора, DL — его длина, N — число, определяющее одно из восьми направлений вектора, CL — его цвет.

CHARF, TEXTF и VALF имеют тот же смысл и такие же форматы вызовов, как и соответствующие подпрограммы CHAR, TEXT и VAL первого пакета. COLOR (X, Y, CL, NBL) предназначена для чтения цвета точки с экрана монитора по ее физическим координатам X и Y; CL — переменная для приема цвета; параметр NBL определяет адрес контроллера в поле адресов ЭВМ. COLOR полезна для различных преобразований изображения на экране, кроме того, являлась базовой подпрограммой для графического редактора. SWAPP предназначена для записи содержимого экрана монитора в дисковый файл. WSHIM и SHIM осуществляют вывод на экран монитора мерцающих точек. Объем второго пакета не превышает 2,5К слов оперативной памяти.

В заключение следует отметить, что приведенный комплекс технических и программных средств может существенно расширить возможности отображения информации при использовании серийно выпускаемых диалоговых вычислительных комплексов на базе микроЭВМ, совместимых с микроЭВМ «Электроника 60».

Телефон для справок: 35-46-42, г. Новосибирск.

ЛИТЕРАТУРА

1. Велихов Е. П., Выставкин А. Н. Проблемы развития работ по автоматизации научных исследований. — М.: ИРЭ АН СССР, 1983. — 52 с. (Препринт / Совет по автоматизации научных исследований при Президиуме АН СССР. Институт радиотехники и электроники; 3 (358)).
2. Нестерихин Ю. Е., Золотухин Ю. Н., Лившиц З. А. Автоматизация: итоги десятилетия. — Автометрия. — 1984. — № 4. — С. 3—14.
3. Гусев О. З., Золотухин Ю. Н., Прохоров О. В., Ян А. П. Базовые конфигурации систем «Микро-КАМАК-лаб». — Автометрия. — 1984. — № 4. — С. 15—20.
4. Зиновьев А. П., Солоненко О. П., Тарасов Б. В. Подсистема цветной машинной графики для микроЭВМ «Электроника 60». — Новосибирск, 1985. — 29 с. (Препринт / ИТФ СО АН СССР; 129—85).
5. СМОГ. I уровень: Инструкция по программированию / Под ред. Ю. А. Кузнецова. — Новосибирск: ВЦ СО АН СССР, 1976. — 78 с.

Статья поступила 23 июля 1986 г.

УДК 681.32

М. Г. Глумов, С. А. Вервер

ДРАЙВЕРЫ ГРАФИЧЕСКИХ УСТРОЙСТВ СМП КУЛОН-1 В ОПЕРАЦИОННОЙ СИСТЕМЕ NTS

Поставляемая с системой машинного проектирования 15-УТ-4-017 (СМП Кулон-1) операционная система «Электроника» позволяет только автоматизировать подготовку и выпуск чертежей и не включает в себя никаких средств разработки нового программного обеспечения. Поэтому в качестве базовой ОС была выбрана операционная система NTS, являющаяся дальнейшим развитием TS-монитора РАФОС А-11. Выбор ОС был обусловлен тем, что она совместима с RT-11 и обеспечивает многопользовательский режим работы. Однако в ней, как и в RT-11, отсутствует поддержка графических устройств СМП КУЛОН-1.

В связи с этим были написаны драйверы всех графических устройств СМП КУЛОН-1, которые позволяют организовать вывод информации на графопостроитель ЭМ-7022 и обмен информацией с рабочим местом проектировщика. В настоящее время передаваемая драйверам информация должна соответствовать системе команд конкретного графического устройства, что очень неудобно. В дальнейших версиях эта информация будет унифицирована. Драйверы написаны на MACRO-11 в соответствии со стандартами ОС, и для включения их в конкретную конфигурацию RT-11 или в NTS необходимо установить ключи условной трансляции в тексте драйвера и оттранслировать его.

Адрес для справок: 44090, г. Ростов-на-Дону, ул. Мильчакова, д. 10. ОКБ «Пьезоприбор», НИО-1. Тел. 24-36-10

Сообщение поступило 8 июля 1986 г.

КРАТКОЕ СООБЩЕНИЕ

Ю. Л. Кетков

Входящая в состав изд-ва «Wirtschaft» (Берлин) редакция по вычислительной технике и обработке данных (ГДР) опубликовала выпуск журнала «EDV-Aspekte» (1986), посвященный описанию входного языка Бэйсик, эксплуатируемого на ПЭВМ Роботрон-1715. Он представляет собой достаточно представительное подмножество расширенной версии Бэйсик-80, разработанной фирмой Microsoft. Для наших читателей входной язык Бэйсик-Роботрон может оказаться полезным, так как с аналогичными версиями приходится работать на ПЭВМ типа MSX, IBM PC, EC-1840.

Изложение сведений по языку завершается примерами трех практических программ. С помощью первой из них «рабочее место технолога» на основании нормативных документов, технологических характеристик изделия и оборудования рассчитывается трудоемкость (временные затраты) на различных этапах обработки при выполнении сверлильных, шлифовальных и резбонарезных операций. Вторая программа Schulmathematik «школьная математика» реализует пошаговое вычисление бесконечных выражений, операндами которых могут быть числовые величины и стандартные функции. Программа «псевдографика» обеспечивает вывод на экран дисплея или на АЦПУ графиков функций, заданных пользователем с помощью оператора DEF FN, линий уровня и диаграмм. При этом имеется возможность нанести координатные оси, сделать их разметку и т. п.

Нами были исследованы некоторые пути модификации программы «школьная математика» в связи с актуальностью разработки лабораторных программ подобного рода и необходимостью накопления соответствующих практических приемов. Анализируемая программа имити-

(Продолжение см. на стр. 86)

УДК 681.322—181.48

Д. А. Страбыкин

ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ С ПОМОЩЬЮ ДИАЛоговых СИСТЕМ УПРАВЛЕНИЯ

Диалоговые системы управления (ДСУ) позволяют повысить эффективность экспериментальных исследований микропроцессорных устройств (МПУ) на базе микроЭВМ. Аппаратные средства ДСУ включают в себя серийное оборудование: «Электронику МС 1201.01», алфавитно-цифровой дисплей, накопитель на гибких магнитных дисках «Электроника 70» и печать УВВП4-30-004, входящие в состав ДВК-2М, а также специально разработанное устройство сопряжения (УС), содержащее набор ПМС, соединенных внутренней магистралью, буферный формирователь (БФ), обеспечивающий гальваническую развязку и электрическое согласование шин этой магистрали с системной магистралью МЭ (рис. 1). Конструктивно УС выполнено в виде отдельного блока с габаритными размерами 550×496×465 мм и встроенным источником питания, содержит до 8 ПМС.

Программное обеспечение ДСУ имеет ряд автономных программируемых модулей, обеспечивающих ввод данных и микрокоманд (МК); проведение исследований в автономном и тактовом режимах; отображение на экране дисплея состояния входов, выходов и внутренних регистров микропрограммируемого устройства. Исследуемое устройство (ИУ) подключается к ПМС через два разъема ГРПМ-61, имеющих по 32 контакта.

Программируемый модуль сопряжения представлен в адресном пространстве микроЭВМ пятью 16-разрядными регистрами и позволяет организовать независимый объем данными с ИУ каждой из 64 линий связи (рис. 2). Принцип действия основан на программном задании уровня сигнала путем управления значениями разрядов в РПД и получении информации с ИУ с помощью программного опроса РПМ. Каждой линии связи с ИУ соответствует разряд в РПД и одним из бу-

ферных регистров, разряд в РПМ и одной из групп (В, С, Д, Е) коммутатора. Буферные регистры, РПД, РПМ и группы коммутатора имеют адреса в адресном пространстве МЭ. Буферные регистры и РПД доступны только для записи, а РПМ и группы коммутатора — для чтения, поэтому одни и те же адреса соответствуют РВ и группе В, РС и группе С, РД и группе Д, РЕ и группе Е. Адреса регистров не являются фиксированными и могут быть смещены в адресном пространстве микроЭВМ при подключении нескольких ПМС.

Особенность операций обмена с РПД и РПМ — данные по СМ не передаются при выполнении команд обращения к этим регистрам. При выполнении команды записи в РПД происходит запись содержимого буферных регистров. Использование РПД обеспечивает синхронность изменения сигналов на входах измерительного прибора (ИП). Операция чтения РПМ заключается в записи информации от ИУ в РПМ. Входы РПМ подключены ко всем контактам разъемов, связывающих ПМС с ИУ, поэтому кроме сигналов от ИУ в него записываются и сигналы, подаваемые на ИУ. Использование РПМ обеспечивает одновременный прием сигнала от исследуемых устройств на всех линиях связи.

При проведении исследований микроЭВМ подает через ПМС на ИУ управляющие и информационные сигналы и принимает ответные сигналы. При подаче сигналов на ИУ микроЭВМ сначала заполняет буферные регистры; на СУ поступает адрес регистра, в который будет записываться данные и сигнал ВЫВОД (запись). Данные с системной магистрали через блок формирователей, ШФ поступают на входы буферных регистров и одним из сигналов синхронизации (СИ1, СИ2, СИ3, СИ4) от СУ заносятся в выбранный регистр. После за-

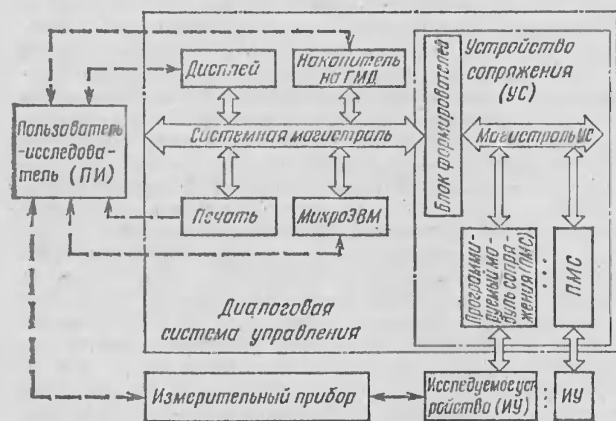


Рис. 1. Структурная схема системы экспериментальных исследований микропроцессорных устройств на базе микроЭВМ

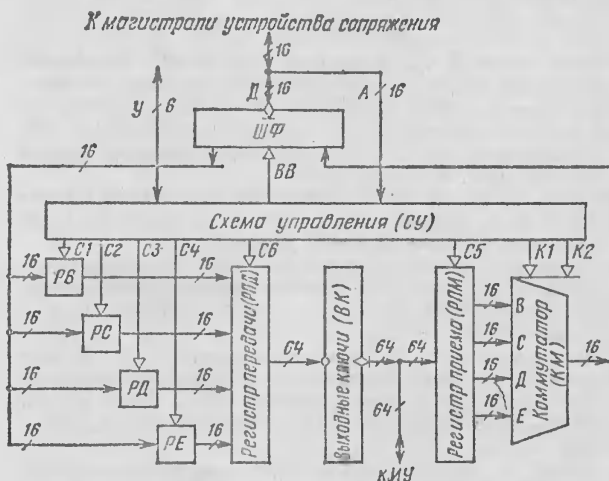


Рис. 2. Функциональная схема программируемого модуля сопряжения

полнения буферных регистров от микроЭВМ поступает команда записи в РПД, происходит пересылка данных из буферных регистров в РПД и изменение сигналов на входах ИУ; устанавливается адрес РПМ и выдается сигнал ВВОД (чтение), по которому сигналы на линиях связи от ИУ запоминаются в регистре РПМ.

Для чтения содержимого РПМ на СУ от микроЭВМ поступает адрес одной из групп КМ и сигнал ВВОД. При этом СУ обеспечивает передачу информации с заданной входной группы КМ на его выход и формирует сигнал (ВВ), с помощью которого данные с выхода КМ передаются на системную магистраль. Аналогично производится чтение информации с других входных групп КМ.

Исследование секционированных МПУ.

Операционное устройство (ОУ) секционированного микропроцессора может быть самостоятельно изготовлено ПИ в виде отдельной платы, подключаемой к программируемому модулю сопряжения, так как для исследования простейшего 8-разрядного ОУ достаточно четырех центральных процессорных элементов К589ИК02 или двух микропроцессорных секций КР1804ВС1.

При исследовании ОУ диалоговая система управления работает в двух режимах: выполнение микрокоманды и ввод группы микрокоманд. В режиме выполнения микрокоманды на экран дисплея выводится форма, содержащая область ввода одной микрокоманды и отражающая состояние ОУ до и после выполнения микрокоманды (рис. 3). В начале работы в режиме выполнения микрокоманд ПИ используя цифровые клавиши, клавиши управления курсором и дисплеем, вводит двоичный код МК в соответствии с форматом области ввода. После записи кода можно или выполнить данную МК, нажимая клавишу Т (ДСУ посылает в ОУ введенный код МК) и тактовый сигнал, или перейти в режим ввода группы микрокоманд. После выполнения МК ДСУ переписывает в левую часть экрана (состояние ОУ до выполнения МК) информацию из правой части (состояние ОУ после выполнения МК) и заполняет правую часть новыми результатами, считанными из ОУ.

В режиме ввода группы микрокоманд на экран дисплея выводится специальная форма, позволяющая одновременно наблюдать до двадцати МК.

При полном заполнении экрана информация автоматически сдвигается вверх на одну строку. Для исследования ОУ в ДСУ предусмотрены ввод и хранение 64 микрокоманд.

Устройство управления с программируемой логикой (УУПЛ) с помощью ДСУ делится на память МК и схемы формирования адреса микрокоманды. Память МК программно моделируется в ДСУ, облегчая ввод и редактирование микропрограмм, упрощается аппаратная часть УУПЛ, содержащая необходимые интегральные микросхемы, отражающие особенности структуры и функционирования исследуемого УУПЛ.

№ КОД МИКРОКОМАНДЫ	ШИНА ДАННЫХ		АДРЕС		АДРЕС		УПРАВЛЯЮЩИЕ СИГНАЛЫ		ОУ МПС К1804ВС1	
	18-16	15-13	12-10	D	A	B	CO	OE		SC1
До выполнения	МИКРОКОМАНДА		ПОСЛЕ ВЫПОЛНЕНИЯ		МИКРОКОМАНДА					
Р0:	ВЫХОДНЫЕ СИГНАЛЫ:		Р0:		ВЫХОДНЫЕ СИГНАЛЫ:					
01		01						
11		11						
21		21						
31		31						
41		41						
51		51						
61		61						
71		71						
81		81						
91		91						
101		101						
111		111						
121		121						
131		131						
141		141						
151		151						

Рис. 3. Форма на экране дисплея для исследования операционного устройства на основе БИС КР1804ВС1

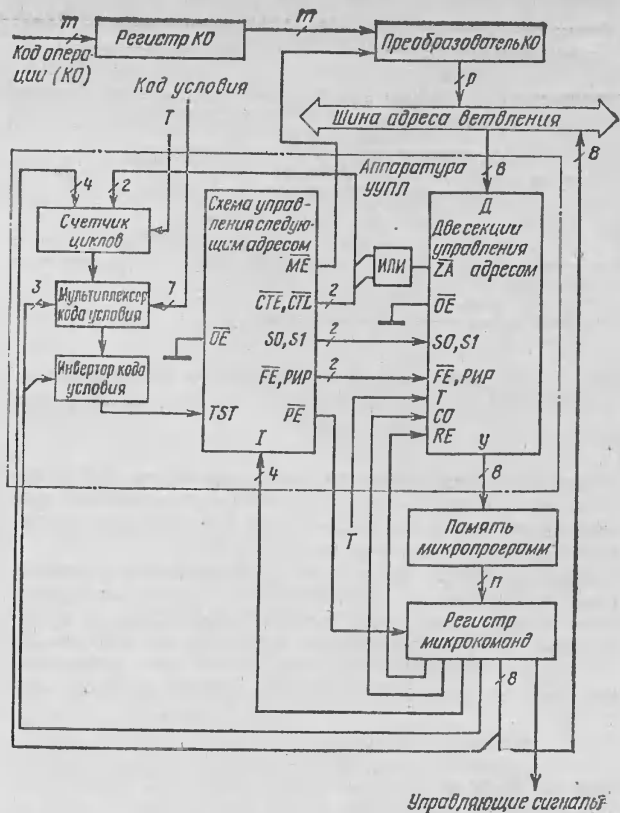


Рис. 4. Структурная схема устройства управления с программируемой логикой на основе БИС серии КР1804

На структурной схеме УУПЛ выделена часть устройства, реализуемая аппаратно на отдельной плате, подключаемой к ПМС (рис. 4). ДСУ применяется также для исследования УУПЛ на основе БИС К589ИК01, КР1804ВУ4.

Микропрограммируемый МП получается соединением ОУ и УУПЛ. ДСУ позволяет использовать для исследования МП платы ОУ и УУПЛ, связь между которыми осуществляется через ПМС программным путем. Для исследования МП требуется память данных и программ, которая так же, как и память микропрограмм УУПЛ, программно моделируется ДСУ.

Исследование однокристалльных МПУ

Исследуемые МПУ содержат БИС КР580ИК80, генератор тактовых импульсов и регистр слова состояния (рис. 5). ДСУ выполняет функции памяти программ и данных, внешних устройств МП, формирует все необходимые ответные сигналы управления. Взаимодействие ДСУ и МПУ организовано на двух уровнях управления МП: микропрограммном и программном. В первом случае в каждом машинном цикле МП проверяет состояние сигнала готовности ГОТ, выдаваемого ДСУ на шину управления. При нулевом состоянии сигнала ГОТ работа МП прекращается и осуществляется копирование состояния выходов МПУ в память ДСУ. Состояния входов и выходов МПУ отображаются на экране дисплея при исследовании работы МП по машинным циклам.

При организации взаимодействия на программном уровне управления ДСУ в первом машинном цикле работы МП при выполнении каждой исследуемой команды подает в МПУ сигнал запроса на прерывание ЗАПР. Если прерывания не запрещены, то после завершения выполнения команды МП переходит к специальной программе обработки прерывания, которая копирует содер-

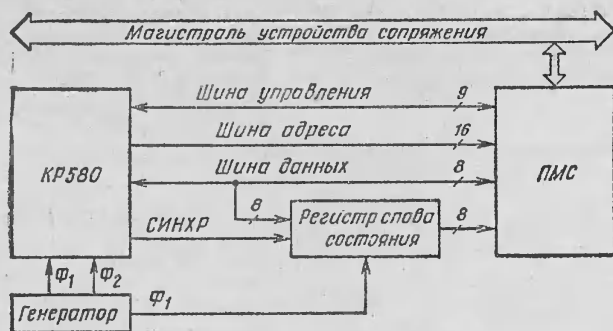


Рис. 5. Подключение однокристалльного микропроцессора КР580ИК80 к программируемому модулю сопряжения

жимое всех программно-доступных регистров МП в память ДСУ. Содержимое внутренних регистров МПУ отображается на экране дисплея при исследовании выполнения команд и отладке программ МП.

Применяя ДСУ, исследуют: программируемое устройство ввода-вывода (КР580ВВ55), программируемый блок приоритетного прерывания (КР580ВН59) и интервальный таймер КР580ВИ53. Представляет интерес исследование микропроцессорных систем под управлением ДСУ с организацией связей между устройствами

УДК 681.323:181.48

А. А. Мячев

МИКРОКОНТРОЛЛЕРЫ ЭКОНОМИЧНЫХ ПОДСИСТЕМ АВТОМАТИЗАЦИИ НА БАЗЕ ЕВРОКАРКАСОВ И ПРИБОРНЫХ МАГИСТРАЛЕЙ

Персональные ЭВМ (ПЭВМ) используются для построения массовых систем автоматизации экспериментов (САЭ) [1, 2]. Широкое внедрение микропроцессоров, материал- и энергосберегающих технических решений, интерфейсных БИС в САЭ стимулировало создание и развитие программно-управляемых модульных подсистем на основе экономичных еврокаркасов (высотой 3U) с магистралями типа IEEE-488 (КОП по ГОСТ 26.003-80) и Q-bus (МПИ по ГОСТ 26.765.51-86) [3, 4].

Значительная часть наиболее экономичных серийных ПЭВМ массового применения оформляется в виде функционально и конструктивно законченных изделий (закрытого типа). Связь с внешним (пользовательским) оборудованием в ПЭВМ осуществляется через унифицированные интерфейсы RS-232C (стык С2), ИРПС, Centronics (ИРПП-М) и лишь в некоторых ПЭВМ — через интерфейс IEEE-488. Системный интерфейс ПЭВМ, как правило, отличается от системных интерфейсов, совместимых с ПЭВМ, по набору команд микроЭВМ (являющихся преимущественно ЭВМ открытого типа), сохраняя основные принципы их организации. При построении САЭ на базе таких ПЭВМ нельзя непосредственно использовать существующие контроллеры приборных интерфейсов (и принципы их построения) для совместимых с ПЭВМ микроЭВМ. Поэтому актуально создание микроконтроллеров, связанных с ПЭВМ через внешние интерфейсы типа стык С2/ИРПС.

Экономичные массовые средства автоматизации научных экспериментов ориентируются на использование ев-

системы через ПМС программным путем. В этом случае обеспечивается доступ к основным управляющим и информационным трактам системы, позволяющим отображать на экране ДСУ состояние входов и выходов и содержимое внутренних регистров устройств, работающих в системе.

Адрес для справок: 610023, г. Киров, ул. Коммуны, д. 36, Политехнический институт, кафедра ЭВМ. Тел. 9-10-70

ЛИТЕРАТУРА

1. Микро- и мини-ЭВМ / Е. П. Балашов, В. Л. Григорьев, Г. А. Петров: Учебное пособие для вузов.— Л.: Энергоатомиздат. 1984.
2. Микропроцессоры: средства отладки, лабораторный практикум и задачник: Учебник для ВТУЗов / Н. В. Воробьев, В. Л. Горбунов, А. В. Горячев и др. Под ред. Л. Н. Преснухина.— М.: Высшая школа, 1986.
3. Проектирование цифровых систем на комплектах микропрограммируемых БИС / С. С. Булгаков, В. М. Мещеряков, В. В. Новоселов, Л. А. Шумилов. Под ред. В. Г. Колесникова.— М.: Радио и связь, 1984.
4. Страбыкин Д. А. Управление цифровыми измерительными приборами с помощью встроенных микро-ЭВМ // Приборы и системы управления.— 1983.— № 11.— С. 11—13.

Статья поступила 2 июня 1986 г.

рокаркасов и внутрикаркасных магистралей типа КОП и МПИ и интерфейсных программно-управляемых БИС, обеспечивающих по сравнению с типовыми программными контроллерами КОП и МПИ расширенные возможности по основным показателям назначения [2—4, 6, 8].

В большинстве ПЭВМ используются микропроцессоры типа К580ИК80А, К1810ВМ86, К1801ВМ, К1811ВМ. Применение в микроконтроллерах КОП основных типов микропроцессоров, ОЗУ, ПЗУ, ППЗУ и программируемых интерфейсных БИС серии К580ВК91/92, матричных кристаллов серии К1801ВП1 обеспечивает развитые функциональные возможности, высокую надежность, малые габаритные размеры и потребляемую мощность, низкую стоимость и высокую технологичность. Для таких микроконтроллеров характерны многообразие режимов программной настройки интерфейсных БИС, в том числе возможность работы в режиме прямого доступа в память (ПДП), и эмуляции серийных контроллеров КОП для микроЭВМ типа СМ1300, СМ1300-01, «Электроника 60», СМ1800, СМ1810 и др., поддерживаемых средствами программирования [8].

В настоящее время в одноплатных микроконтроллерах можно использовать типовые унифицированные базовые фрагменты основных функциональных блоков [5]: центрального микропроцессора с ОЗУ, ПЗУ/ППЗУ ограниченного объема, интерфейсного блока КОП, стыка С2/ИРПС, автозагрузчика, ПЗУ/ППЗУ пользователя. Для фрагмента микропроцессора можно применять совместимые одноплатные микроЭВМ, имеющие необходимый объем ОЗУ и ПЗУ, а также обеспечивающие быструю реакцию на внешние события и связь с ЭВМ. Для более эффективного применения микроконтроллеров целесообразно предусмотреть установку дополнительных БИС ПЗУ/ППЗУ в свободные розетки, расположенные во фрагменте пользователя.

Исследования показали, что в зависимости от архитектуры ПЭВМ закрытого типа и требуемой производительности микроконтроллера более экономично иметь микроконтроллеры двух типов по архитектуре используемого МП и двух классов по производительности МП. Основные характеристики микроконтроллеров КОП,

Характеристики микроконтроллеров КОП

Фрагменты блоков	Тип микроконтроллера			
	Первый		Второй	
	СП	ВП	СП	ВП
Блок центрального МП: тип МП объем ОЗУ, К байт объем ПЗУ/ППЗУ, К байт	K1801BM1 8 4	K1801BM2 8...16 4...8	K1821BM85 8 4	K1810BM86 8...16 4...8
Блок КОП: тип БИС КОП число каналов КОП режим прямого доступа	K1801BП1 1 —	KP580 1...2 +	KP580 1 +	KP580 1...2 +
Блок стык С2/ИРПС: число каналов тип БИС	1	1...2	1	1...2
Блок периферийного МП: тип МП тип БИС КОП	—	K1801BП1-035 K1821BM85 KP580	—	KP580 B51 K1821BM85 KP580
Система программирования ассемблер (А), Бейсик (Б), Фортран (Ф), Паскаль (П)	А, Б, Ф	А, Ф, П	А, Б, Ф	А, Ф, П

Примечание. СП — средней производительности, ВП — высокой производительности.

сгруппированные по фрагментам блоков, приведены в таблице.

В перспективный вариант фрагмента блока КОП входят две интерфейсные БИС К580ВК91 и К580В192, два 8-разрядных приемопередатчика КОП и дополнительно устанавливаемая БИС канала ПДП, которая управляется резидентными программами, размещенными в ОЗУ (ППЗУ).

Фрагмент блока стык С2/ИРПС в зависимости от применяемого микропроцессора содержит определенные типы программируемых БИС (KP580BV51, K1801BП1-035, K581BA1), а также дополнительные микросхемы серии K170, осуществляющие прямое и обратное преобразование уровней сигнала TTL/стык С2 (в минимальном варианте по четырем целям стыка 103...106).

Экономичные варианты дополнительных базовых операционных систем реального времени являются полностью резидентными, содержат драйверы контроллера КОП, обеспечивают работу с таймером, обслуживание прерываний, загрузку части системы и обмен информацией через блок стыка С2/ИРПС с ЭВМ.

Прикладное программное обеспечение (ПО) микроконтроллеров базируется на унифицированных для ПЭВМ языках высокого уровня (Бейсик, Фортран-77, Паскаль и др.) и разрабатывается в среде унифицированных операционных систем на типовых инструментальных ПЭВМ (или микроЭВМ).

Для САЭ, построенных на серийных приборах и устройствах, выходящих на КОП, возможны реализации микроконтроллеров в виде автономных унифицированных блоков с типовой платой микроконтроллера, блоком питания и разъемами для подключения к ЭВМ по последовательному каналу и кабелями магистрали КОП. В сложных лабораторных САЭ коллективно используются несколько ПЭВМ с большим числом приборов и устройств, выходящих на КОП. Для таких САЭ целесообразно реализовать автономные микроконтроллеры с локальной сетью ПЭВМ (в перспективе и на базе последовательного канала КОП с соответствующим ПО локальной сети [6, 8]).

В микроконтроллерах перспективных быстродействующих 16-разрядных магистралей (типа КОП) еврокаркасов высотой 6U возможно совместное использование микропроцессоров типа K1801BM2, K1810BM86, однокристалльных микроЭВМ типа K1816VE48/51 в различных сочетаниях в зависимости от требований САЭ к

объему и сложности первичной обработки данных и реакции на внешние события.

Комплексная реализация программы создания экономических средств автоматизации на базе еврокаркасов и магистрали КОП обеспечит построение массовых САЭ на базе ПЭВМ, в том числе персональных автоматизированных рабочих мест для испытания и метрологического контроля промышленной продукции [7], типовых учебно-исследовательских лабораторий и тренажеров, рабочих мест операторов и технологов гибких производственных систем, исследователей при проведении несложных лабораторных и инженерных экспериментов в самых широких областях науки и техники.

Телефон для справок: 238-40-45, Москва.

ЛИТЕРАТУРА

1. Громов Р. Г. Профессиональные приложения персональных ЭВМ // Микропроцессорные средства и системы.—1985.— № 3. С. 9—15.
2. Перспективы развития персональных компьютеров и их использование в задачах автоматизации измерений / Б. Н. Наумов, В. Г. Захаров, А. А. Мячев и др.— В кн.: V Всесоюзный симпозиум по модульным информационно-вычислительным системам.— Кишинев: Штиинца, 1985.— С. 117—118.
3. Colett R. Recent Developments in Data Acquisition Systems // Electronics Test.—1984.— Feb.— P. 53—59.
4. Cleaveland P. What's new in data acquisition? // The Industrial and Process Control Magazine.—1984.— May.— P. 29—34.
5. Гальперин М. П. Одноплатные микроЭВМ и микроконтроллеры // Микропроцессорные средства и системы.—1984.— № 2.— С. 16—19.
6. Мячев А. А., Никольский О. А. Стандартные интерфейсы микропроцессорных систем // Микропроцессорные средства и системы.—1984.— № 1.— С. 27—33.
7. Интерфейс для программируемых приборов — основа систем автоматизации метрологических испытаний / И. М. Грязнов, Л. С. Ситников и др.— В кн.: V Всесоюзный симпозиум по модульным информационно-вычислительным системам.— Кишинев: Штиинца, 1985.— С. 70—71.
8. Виноградов В. И. Информационно-вычислительные системы: Распределенные модульные системы автоматизации.— М.: Энергоатомиздат, 1986.— 336 с.

Статья поступила 24 декабря 1985 г.

Краткое сообщение

АЦП, предназначенные для преобразования постоянного напряжения в параллельный двоичный код, в основном применяются в системах автоматизации научных исследований и промышленных испытаний, komponуемых на основе крейтов КАМАК и измерительно-вычислительных комплексов.

Поставка по нарядам ВО «Союзглавприбор» через территориальные УМТС.

Адрес: 428000, г. Чебоксары, пр. И. Яковлева, 3, Чебоксарское ПО «Электроприбор».

Телефоны: 21-93-79, 21-93-15, отдел сбыта — 29-75-04.

Кроме того, возможно, многим читателям журнала будет интересно узнать, что Чебоксарское ПО «Электроприбор» выпускает универсальный бытовой деревообрабатывающий станок модели СУБД.

Станок предназначен для обработки пиломатериалов в быту (распиловки древесины вдоль и поперек волокон, фугования, сверления, фрезерования, а также для заточки инструмента). Розничная цена 343 руб.

Адрес изготовителя: 428000, Чебоксары, пр. И. Яковлева, 3.

Технические характеристики АЦП

Наименование показателя	ФК4224	ФК4225	ФК4226
Диапазон преобразования входного сигнала, В	$\pm 4,096$ $\pm 2,048$ $\pm 1,024$ $\pm 0,512$	$\pm 1,024$ $\pm 0,512$ $\pm 0,256$	$\pm 1,024$ $\pm 0,512$ $\pm 0,256$
Количество двоичных разрядов, вкл. знак	67	10	8
Предел допускаемой систематической составляющей погрешности, %	5	0,4	2
Ступень квантования, мВ	8	2	—
Максимальная частота тактирования, МГц	50	1	20
Время преобразования, мкс	0,1	1,2	0,25
Время цикла кодирования, мкс	0,02	1,0	0,05
Входное сопротивление, Ом	$50 \pm 3,75$	$50 \pm 3,75$ $100 000 \pm 1000$	75 ± 2
Напряжение питания, В	$\pm 24, \pm 6$	$\pm 24, \pm 6$	$\pm 24, \pm 6$
Объем ОЗУ, К байт	4	4	1
Число занимаемых станций	2	1	1
Наработка на отказ, ч	12 000	12 000	12 000

УДК 621.316.722.1

И. И. Антоняк, Я. С. Зайшлый, А. Е. Фриш

ИМПУЛЬСНЫЙ СТАБИЛИЗАТОР СПЕЦИАЛИЗИРОВАННЫХ НАПРЯЖЕНИЙ ДЛЯ МИКРОПРОЦЕССОРА КР580ВМ80

Данный импульсный стабилизатор специализированных напряжений для управляющих систем построен на основе микропроцессора КР580ВМ80.

Предлагаемая схема (см. рисунок) позволяет отказаться от применения дополнительных стабилизаторов, имеет три функциональных блока: генератор, формирователь напряжения -5 В и формирователь напряжения $+12$ В.

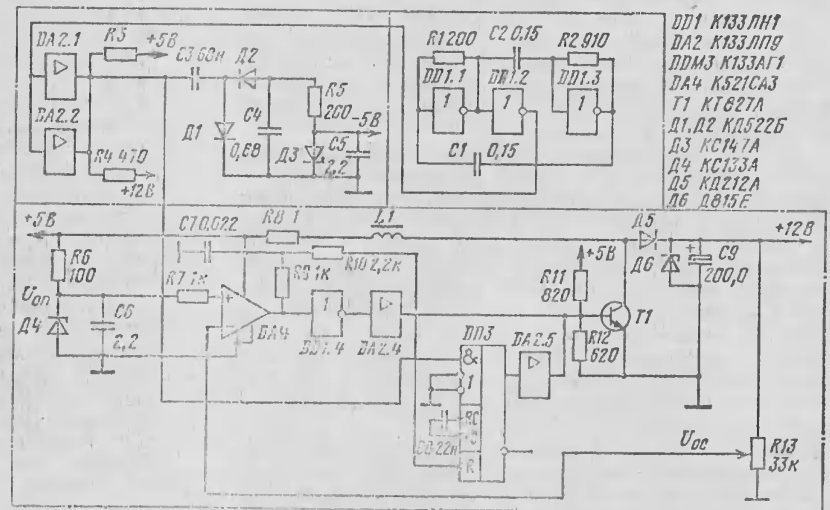
На микросхеме DD1 собран автогенератор импульсов частотой порядка 10 кГц, задаваемый элементами R1, R2, C1, C2.

Формирователь напряжения -5 В реализован по схеме двух дублированных бестрансформаторных преобразователей постоянного напряжения, отдельно питающихся от источника $+5$ В, и формирователя $+12$ В. Прямоугольные импульсы с выходов буферных повторителей DA2.1 и DA2.2 выпрямляются и сглаживаются через D1, D2, C3, C4, D5, D6, C9. Параметрический стабилизатор, построенный на R5, D3, C5, обеспечивает стабилизацию напряжения -5 В.

Формирователь напряжения $+12$ В реализован по схеме импульсного релеяного регулятора с последовательным включением дросселя и параллельным включением транзистора.

Прямоугольные импульсы подаются на выход одновибратора DD3, который формирует импульс длительностью 60 мкс, задаваемый элементами R10 и C8. С выхода одновибратора через буферный повторитель DA2.3

импульсы поступают на базу коммутирующего транзистора T1. Электродвижущая сила самоиндукции, возникающая в дросселе L1 в моменты закрытия транзистора T1, вызывает возрастание его коллекторного напряжения и заряд конденсатора C9 через блокирующий диод D5, предотвращающий разряд конденсатора C9 в моменты пребывания транзистора T1 в открытом состоянии. Стабилизатор D6 обеспечивает защиту от превышения выходного напряжения. Стабилизация выходного напряжения обеспечивается путем сравнения ком-



Предложенная схема импульсного стабилизатора специализированных напряжений

паратор DA4 напряжения $U_{ос}$, подаваемого через резистор R13, и опорного напряжения $U_{оп}$, формируемого параметрическим стабилизатором D4, R6, C6. При превышении опор-

ного напряжения одновибратор DD3 блокируется в нулевом состоянии, что приводит к прекращению тока в дросселе до наступления момента сравнимых величин $U_{ос}$ и $U_{оп}$. Ре-

зистор R9 обеспечивает работу компаратора DA4 в режиме однополярного питания.

Статья поступила 24 ноября 1988 г.

УДК 62.52 : 621.387

В. С. Поздеев, О. Н. Тихонов

ОТОБРАЖЕНИЕ ИНФОРМАЦИИ НА МАТРИЧНОМ ЖИДКОКРИСТАЛЛИЧЕСКОМ ДИСПЛЕЕ ИЖВ

Данное устройство отображения предназначено для визуализации алфавитно-цифровой и графической

информации в вычислительных системах с интерфейсом «Электроника 60», например микроЭВМ на ба-

зе микропроцессорного комплекта серии K588 [1] или контроллера на базе однокристального микропроцессора K1801BM1 [2].

Устройство отображения на матричном жидкокристаллическом дисплее ИЖВ 71=96×8 (рис. 1) содержит генератор импульсов (ГИ), жидкокристаллический дисплей (ЖКД), коммутатор (КМ), компаратор (КП), магистральный приемник (МП), ОЗУ,

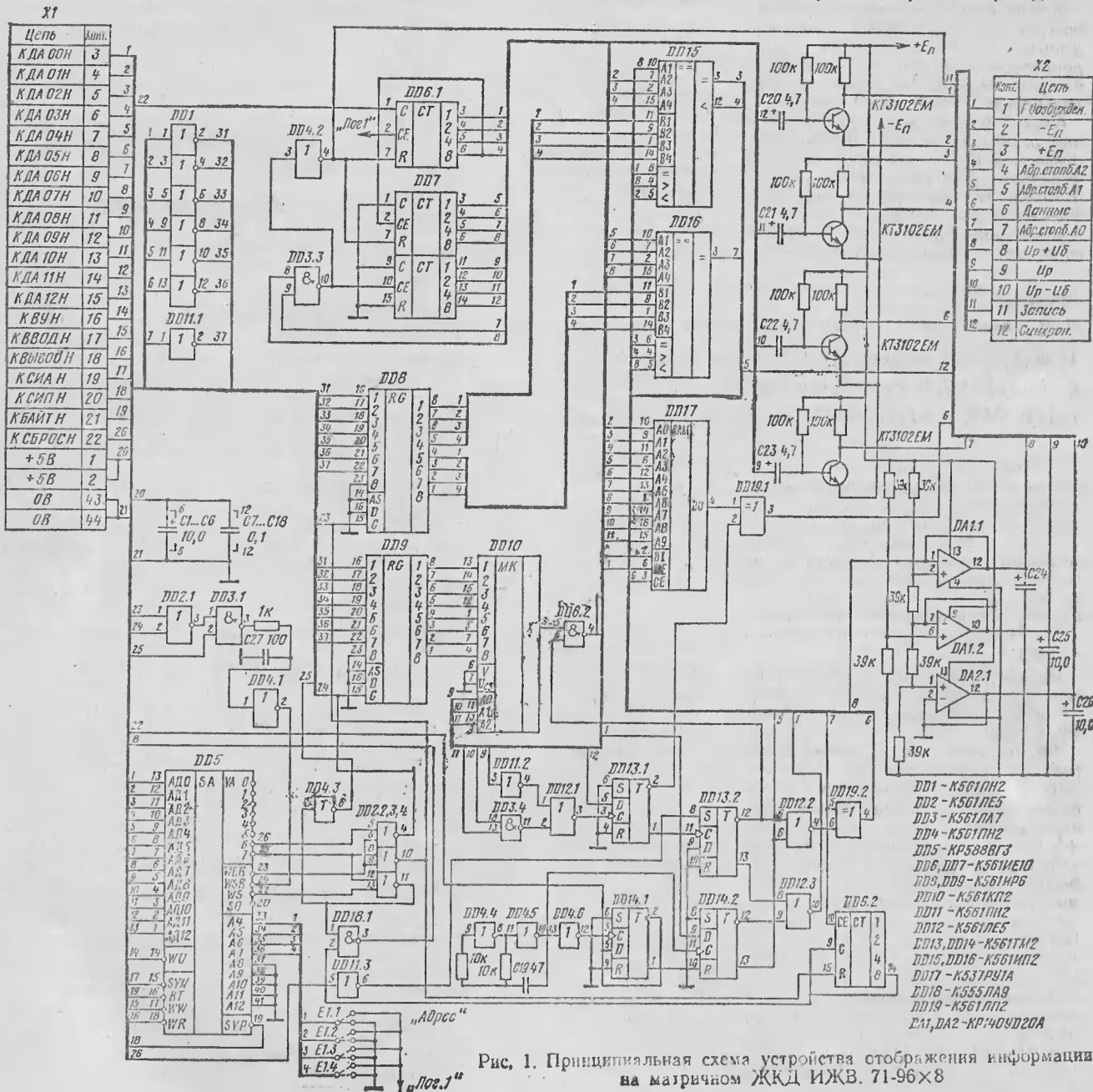


Рис. 1. Принципиальная схема устройства отображения информации на матричном ЖКД ИЖВ. 71-96×8

регистр адреса (РА), регистр данных (РД), селектор адреса (СА), счетчик столбцов (СТБ), счетчик задержки (СТЗ), счетчик строк (СТР) и формирователь опроса (ФРО).

Адреса регистров 177000 ... 177776 устанавливаются с помощью переключек на плате устройства отображения. Запись адреса и данных столбцов жидкокристаллического дисплея в регистры осуществляется по сигналам Выв. РА и Выв. РД, формируемым в селекторе адреса при совпадении синхроимпульса адреса К СИА Н, сигнала К ВЫВОД Н и адресов регистров.

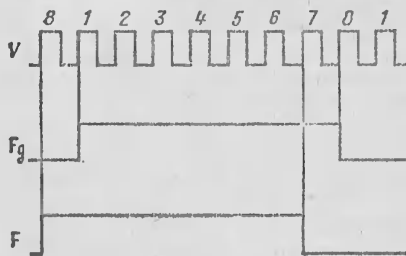
При поступлении импульса генератора содержимое счетчика столбцов увеличивается на единицу (рис. 2). Восемьразрядный выходной код СТБ является частью адреса ОЗУ и поступает на вторую группу входов КП. При достижении СТБ значения 96 на его выходе формируется импульс, увеличивающий содержимое счетчика строк и одновременно поступающий на управляющий вход ЖКД. Три младших разряда СТР определяют адрес строки, а старший разряд счетчика является сигналом возбуждения ЖКД. По положительному фронту сигнала на стробирующем входе ЖКД входная информация записывается во входной регистр



2. Временная диаграмма импульсов, формируемых генератором:

а — синхронизирующие импульсы КП, СТЗ и ЖКД;
б — синхронизирующие импульсы СТБ и импульсы управления

дисплея, а по отрицательному фронту на управляющем входе — переписывается в выходной регистр, который непосредственно управляет столбцами дисплея (рис. 3). Запись данных столбцов в ОЗУ осуществляется при равенстве содержимого счетчика столбцов записи в регистре адреса. В этом случае на выходе КП при совпадении импульса генератора и сигнала Выв. РД формируется сигнал



3. Временные диаграммы управляющих сигналов ЖКД

записи в ОЗУ. На информационный вход ОЗУ, при изменении содержимого счетчика строк, с выхода комму-

татора последовательно поступает содержимое регистра данных. Одновременно изменяется содержимое счет-

```

1 ; АССЕМБЛЕР Э-60
2 ;*****
3 ;ПРОГРАММА ИНДИКАЦИИ ЦИФРОВЫХ СИМВОЛОВ
4 ;*****
5 .TITLE FORM
6 .CSECT
7 .GLOBE RAS,RDS,FORM,MKS,KS
8 RAS=177760 ;РЕГИСТР АДРЕСА
9 RDS=177762 ;РЕГИСТР ДАННЫХ
10 ;%0-РЕГИСТР КОДА СИМВОЛА
11 ;MKS-АДРЕС МАССИВА СИМВОЛОВ
12 ;KS-ПОЗИЦИЯ СИМВОЛА (0-95)
13 ;SCS-СЧЕТЧИК СТОЛБЦА СИМВОЛА
14 00000 010246 FORM: MOV #2,-(6)
15 00002 010146 MOV #1,-(6)
16 00004 162700 SUB #6,%0
17 00010 005700 TST %0 ;АНАЛИЗ КОДА СИМВОЛА
18 00012 100410 BMI STOP
19 00014 020027 CMP %0,#9.
20 00020 003005 BGT STOP
21 00022 004767 JSR #7,M1
22 00026 062701 ADD #MKS,%1 ;ФОРМИРОВАНИЕ АДРЕСА СИМВОЛА
23 00032 000407 BR M2
24 00034 000000 STOP: HALT
25 ;ФОРМИРОВАНИЕ ПРИРАЩЕНИЯ АДРЕСА СИМВОЛА
26 00036 000000 M1: MOVB: %0,%1
27 00040 000001 ASL #1
28 00042 006301 ASL #1
29 00044 006301 ADD #0,%1
30 00046 060001 RTS #7
31 ;ФОРМИРОВАНИЕ ПОЗИЦИИ СИМВОЛА
32 00052 016700 M2: MOV KS,%0
33 00056 006300 ASL %0
34 00060 110067 MOVB %0,KS
35 00064 006300 ASL %0
36 00066 060067 ADD %0,KS
37 00072 005067 CLR SCS
38 ;ФОРМИРОВАНИЕ СИМВОЛА
39 00076 112102 M3: MOVB (1)+,%2
40 00100 005267 INC SCS
41 00104 000074 JSR #7,M4
42 00110 026727 M4: CMP SCS,#5
43 00064 000005
44 00116 100767 BMI M3
45 00120 005002 CLR #2
46 00122 004767 JSR #7,M4
47 00126 005267 INC KS
48 00132 012601 MOV (6)+,%1
49 00134 012602 MOV (6)+,%2
50 00140 026727 M4: CMP KS,#95. ;АНАЛИЗ СТОЛБЦА СИМВОЛА
51 00146 003332 BGT STOP
52 00150 005102 COM #2
53 00152 010267 MOV #2,RAS ;ЗАПИСЬ СТОЛБЦА СИМВОЛА
54 00156 016767 MOV KS,RDS ;ЗАПИСЬ КОДА СИМВОЛА
55 00164 105767 M5: TSTB RAS ;ОЖИДАНИЕ ГОТОВНОСТИ ЛНИ
56 00170 100375 BPL M5
57 00172 005267 INC KS
58 00176 000207 RTS #7
59 00200 000000 SCS: .WORD 0
60 00202 000000 KS: .WORD 0
61 000001 .END

```

чка задержки, выходной сигнал которого является сигналом готовности, и вводится в вычислительную систему по сигналу Вв. РД. Формирователь опроса и элемент «исключающее ИЛИ» выполняют функцию фазовращателя информации.

Формирование символов цифр (от 0 до 9) на дисплее ИЖВ 71=96×8 (программа FORM) осуществляется при последовательном увеличении адреса столбца и занесении соответствующего кода символа в регистр данных. Это определяет начертание символа на дисплее в формате 5×7 элементов. Адреса регистров адреса RAS и данных RDS столбцов равны 177760 и 177762 соответственно. Индикация любого символа на про-

извольном знакоместе инициируется записью его кода в нулевой регистр (%0) и позиции его первого столбца в ячейку KS. В программе возможно автоматическое последовательное формирование символов с любого знакоместа при задании его кода в ячейку KS; анализируется входной код символа и номер задаваемого столбца (содержимое ячейки KS). Коды цифр (0...9) соответствуют ГОСТ 13052—74. При запрещенном коде символа и превышении максимально допустимого номера столбца, равно 96, выполнение программы прекращается.

Справка по телефону: 23-86-07 (г. Ижевск). Поздеев Владимир Степанович.

ЛИТЕРАТУРА

1. Черняковский Д. Н., Шиллер В. А., Юровский А. А. Процессор с системой команд и интерфейсом микроЭВМ «Электроника 60» на основе БИС серии КР588 // Электронная промышленность. — 1983. — № 9. — С. 11—13.
2. Филиппычев С. А., Майдыковский И. В., Борщенко Ю. И., Зубов Ю. В. Применение однокристального микропроцессора К1801ВМ1 в автономных системах сбора и обработки информации // Микропроцессорные средства и системы. — 1985. — № 1. — С. 51—57.

Статья поступила 4 ноября 1986 г.

УДК 621.311.6

А. В. Коломиец, В. К. Горемыкин

ЗАЩИТА ИНТЕГРАЛЬНЫХ СХЕМ ПО ЦЕПЯМ ПИТАНИЯ

При эксплуатации готовых и особенно при ремонте или настройке устройств на базе интегральных микросхем (ИС) напряжение по цепям питания часто превышает максимально допустимое.

При отказе стабилизатора в блоке питания (из-за пробоя регулирующего транзистора, обрыва цепи стабилизатора и т. п.), ошибке при подключении, небрежности и проч. напряжение питания увеличивается в 1, 2...3 раза от установленного значения. Это практически для всех (особенно цифровых) ИС недопустимо.

Простой, применимый при эксплуатации и разработке устройств, метод защиты ИС по цепям питания от перенапряжения и переплюсовки — между выводом питания ИС и общим проводом установить полупроводниковый стабилизатор (в обратном включении) с напряжением стабилизации, равным максимально допустимой величине питающего напряжения. При увеличении напряжения питания до напряжения стабилизации стабилизатора он «пробивается», сохраняя напряжение на выводах питания ИС равным напряжению стабилизации. Резкое возрастание потребляемого тока от источника питания обычно приводит к его отключению по пере-

грузке или к перегоранию защитного плавкого предохранителя. Однако в любом случае напряжение питания на выводах защищаемых ИС не превышает максимально допустимой величины.

Опыт показал, что практически все стабилизаторы (особенно в металлическом корпусе) хорошо выдерживают перегрузку по току и в худшем случае из-за тепловой перегрузки происходит необратимый «пробой» его р-п-перехода, т. е. стабилизатор выходит из строя, но ИС защищается надежно. Всегда дешевле заменить один стабилизатор, нежели менять вышедший из строя целый ряд ИС.

При переплюсовке полярности напряжения питания ИС стабилизатор начинает работать на прямой ветви вольт-амперной характеристики как обыкновенный диод. Падение напряжения на нем не превышает 0,8 В (обычно допустимо для защищаемых ИС).

Пример. Авторами при защите микросхем, выполненных по ТТЛ- и ТТЛШ-технологии (напряжение питания 5 В), используется стабилизатор КС156А. При использовании симметричных стабилизаторов типа КС162.., КС213 параллельно им необходимо ставить обыкновенный мощный диод в обратном включении (или стабилизатор с большим напряжением стабилизации), так как симметричные стабилизаторы не защищают от переплюсовки. При питании ИС (например, КР580ИК80А, К565РУЗ и др.) от нескольких источников напряжения защиту целесообразно предусмотреть по каждому из источников.

Статья поступила 15 июля 1986 г.

НА КНИЖНОЙ ПОЛКЕ

Липаев В. В., Потапов А. И. Оценка затрат на разработку программных средств. — М.: Финансы и статистика, 1988 (1 кв.). 15 л., ил. — В пер.: 1 р. 10 к.

Стоимость разработки и сопровождения программных средств в стране уже сейчас оценивается на уровне миллиардов рублей и довольно быстро прогрессирует. Процесс создания программ для ЭВМ превращается в индустриальный вид человеческой деятельности. Поэтому необходимо развитие экономики этого нового вида индустрии, которая позволит рациональнее использовать капиталовложения и может дать значительный экономический эффект. В монографии систематически изложены методы проведения технико-экономического анализа процесса разработки программных средств. Базой оценки затрат и влияния на них различных факторов являются исследования реальных технологических процессов и технико-экономических характеристик разработки программ. Значительная часть книги является оригинальной и базируется на исследованиях авторов.

Подробно рассмотрены составляющие затрат на раз-

работку программных средств, а также проведен фрагментный анализ затрат на эксплуатацию и сопровождение программ. Описаны методы сбора и обработки первичных технико-экономических данных о разработках программных средств. Приводятся сведения о фактической трудоемкости, ее распределении по этапам разработки, производительности труда при разработке программных средств для больших универсальных ЭВМ и микроЭВМ, используемых в робототехнических комплексах, ГАПах, при управлении объектами и технологическими процессами. Представлены результаты исследования зависимости длительности разработки от объема комплексов программ и трудоемкости их создания.

Значительное внимание уделено анализу технико-экономической эффективности освоения технологий системами автоматизации разработки программ и средств автоматизированной техники. Даны практические рекомендации по использованию полученных результатов для прогнозирования затрат ресурсов, планирования разработки программных средств и формирования нормативной базы процесса разработки.

С. Д. Чабан, Ю. А. Фильцагин

МОДУЛЬ ДЕСЯТИКАНАЛЬНОГО АНАЛОГОВОГО ВЫВОДА НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Предлагаемый модуль аналогового вывода (рис. 1) обеспечивает преобразование и вывод информации о данных эксперимента по десяти каналам, имеет разрешающую способность 12 двоичных разрядов (старший разряд — знаковый) и оптронную развязку от цифровой части, предназначен для использования в качестве согласующего устройства между микроЭВМ (входящей в систему физического эксперимента) и регистрирующими приборами (осциллографами, самописцами, графопостроителями).

Адреса регистров данных выставляются пользователем с помощью переключек. Формат слова данных приведен на рис. 2. Интерфейсная часть модуля реализует программный режим работы микроЭВМ «Электроника 60». В режиме чтения (цикл ввода) по всем адресам устройство управления (УУ) формирует синхрипульс К СИП Н, который поступает в ЭВМ и подтверждает наличие модуля.

Для вывода цифровой информации процессор иницирует цикл ВВБД. По сигналу К СИА Н срабатывает селектор адреса (СА) и производится запись в регистр адреса (РА). Селектор адреса создает сигнал РАЗР, подаваемый на УУ. Устройство управления при получении К ВВВОД Н запускается и выработывает ЗП и К СИП Н. По сигналу ЗП стробируется дешифратор адреса (ДША) и записывается цифровая информация в регистр данных, которая преобразуется ЦАП в аналоговое напряжение, подаваемое на выходной разъем.

В модуле аналогового вывода предусмотрена возможность перевода в состояние, при котором на всех десяти выходах устанавливается нулевое напряжение с точностью до половины младшего разряда, что обеспечивается записью (100000) в регистры данных. Эта функция реализуется при поступлении сигнала К ПИТН В системной шины (СИШ) ЭВМ, который вырабатывается в случае нарушения сетевого питания или при включении.

Вторичный источник питания (ВИП) (рис. 3) предназначен для питания части модуля, гальванически развязанной от микроЭВМ, представляет собой источник преобразовательного типа «модулятор-демодулятор» и включает в себя преобразова-

тель напряжения, три выпрямителя, три стабилизатора напряжения.

С помощью преобразователя напряжения, выполненного на ИМС

K170AA7, транзисторах КТ816Г и импульсном трансформаторе Т1, постоянное напряжение СШ (12 В) преобразуется в меандр с частотой 25 кГц. С целью ликвидации «сквозных токов», а также уменьшения коммутационных помех в моменты переключения транзисторов формируется «беспокая пауза» с помощью подачи стробирующих импульсов на вход ВК ИМС K170AA7. Импульсный трансформатор выполнен на базе тороидального ферритового сердечника типа М2000НМ-1Б размерами 20×12×6 (обмотки трансформатора

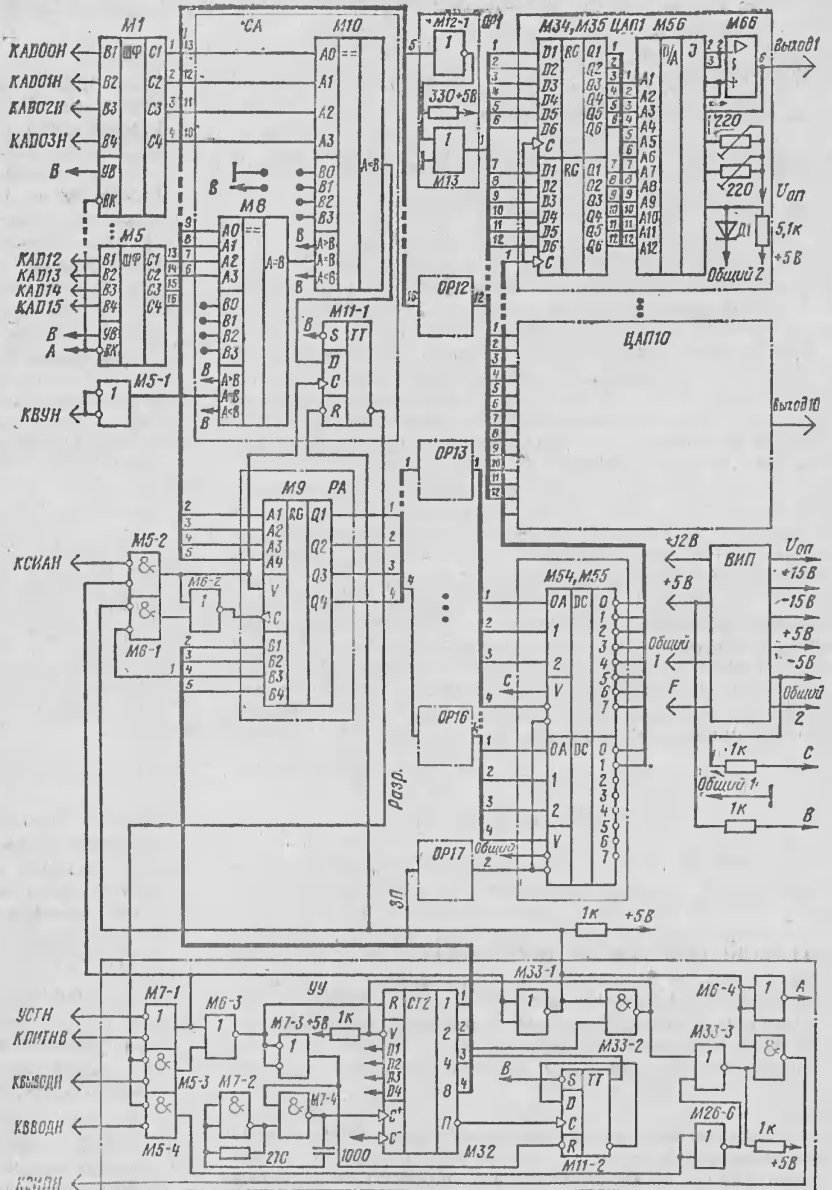


Рис. 1. Принципиальная схема модуля аналогового вывода:

M1...M4—K589A1113; M5 M6—K590P121; M7—K553JA3; M8, M10—K555CP1; M9—K555KP113; M11—K555FM2; M12 M19, M26—K153LH3; M13, M16, M20...M25, M27...M31—K262K1113; M32—K555HE7; M33—K559A1111; M34...M36—K555FM9; M37, M55—K555YD7; M56...M58—59A1A1; M66...M75—140YD6A; D1...D10—KD322B

AD15	AD14	AD13	AD12	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0				

Рис. 2. Формат слова данных модуля аналогового вывода

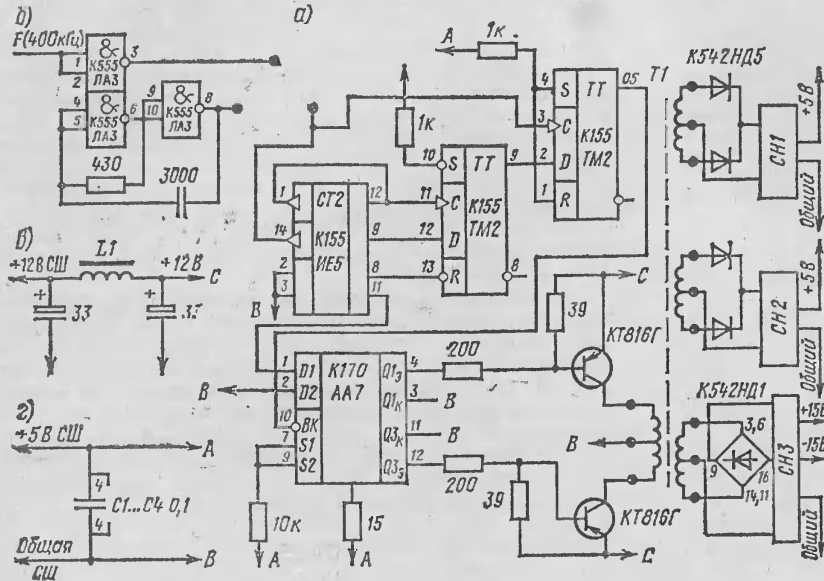


Рис. 3. Вторичный источник питания:

а — схема электрическая принципиальная вторичного источника питания; б — внутренний генератор управляющей частоты; в — фильтр низкой частоты; г — развязывающий конденсатор

тора намотаны из расчета 3 витка на 1 В), выпрямители вторичного напряжения выполнены на диодных сборках К542НД5 и К542НД3, СН1 и СН2 на микросхемах К142ЕН3. Стабилизатор СН3 — компенсационного типа на основе операционных усилителей К140УД6 и регулирующих транзисторов КТ972 и КТ973.

Управляющая частота для преобразователя напряжения 400 кГц подается с СШ или от внутреннего генератора на ИМС К555ЛА3. С целью уменьшения влияния коммутационных помех на СШ и модуль, +12 В СШ подается на преобразователь через П-образный фильтр низкой частоты на основе дросселя L1, который

сделан на тороидальном сердечнике типа М3000НМ размером 16×8×6 и танталовых конденсаторах типа К52-1Б. Обмотка дросселя содержит 30 витков и намотана проводом диаметром 0,7 мм.

Технические характеристики модуля:

Напряжение питания, В	+5, +12
Потребляемый ток, А, не более	0,3, 1,3
Число входных каналов	10
Диапазон выходного напряжения, В	+10
Время установки аналогового напряжения, мкс, не более	12
Разрешающая способность, двоичные разряды (старший-знаковый)	12

Инструментальная погрешность модуля определяется качеством используемых компонентов и их температурными коэффициентами, суммарная погрешность тракта оказывается не более 0,029 %, что меньше шага квантования 11-разрядного ЦАП (погрешность опорного источника — 0,005 % полной шкалы, погрешность ЦАП 594ПА1 — 0,024 % полной шкалы).

Разработанный модуль выполнен на полной плате микроЭВМ «Электроника 60». Низковольтные провода имеют минимальную длину, «аналоговая земля» занимает максимальную поверхность платы, у каждого вывода питания всех аналоговых элементов поставлен развязывающий конденсатор.

Модуль может быть использован в системах для физических исследований, а также в информационно-измерительных и управляющих вычислительных комплексах.

За справками о поставке комплекта конструкторской документации и фотооригиналов печатных плат обращаться по адресу: 614600, Пермь, ГСП, ул. Попова, 9, Центр Научно-технической информации.

Статья поступила 5 декабря 1985 г.

УДК 681.325.5

Р. М. Гнатив, А. Я. Скобылко

ЦИФРОВОЙ ГЕНЕРАТОР СИНУСОИДАЛЬНЫХ СИГНАЛОВ С УПРАВЛЕНИЕМ ПО ЧАСТОТЕ НА СПЕЦПРОЦЕССОРЕ

Предлагаемый цифровой генератор на БИС спецпроцессора «Рената» используется для наладки и тестирования аппаратуры, обработки низкочастотных сигналов.

Однокристалльный процессор цифровой обработки сигналов (ПЦОС) «Рената» создавался для применения в системах, действующих в реальном масштабе времени, поэтому работает с постоянной частотой дискретизации входных непрерывных сигналов. В процессоре заложен

принцип конвейерной обработки данных по командам, хранящимся в памяти, что позволяет выполнять любую команду за определенный промежуток времени, который при тактовой частоте 5 МГц составляет 800 нс [1, 2].

Функциональная схема кристалла ПЦОС «Рената» представлена на рис. 1.

Основные этапы процесса генерации сигнала синусоидальной формы:

генерация напряжения пилообразной формы; изменение формы напряжения до получения трапециoidalного сигнала с вершиной среза 60° (такой сигнал имеет минимизированную 3-ю гармонику).

Далее для получения синусоидального напряжения используется фильтр низкой частоты. При соотношении частоты дискретизации и генерируемой частоты порядка 20..30 достаточно использовать фильтр 2-го порядка на одном операционном усилителе [3]. При меньшем соотношении требуется более сложный фильтр. Характеристики фильтра должны быть согласованы с частотным диапазоном генератора.

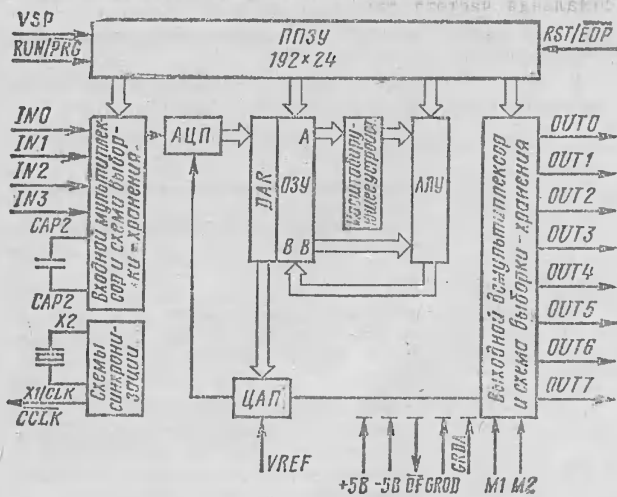


Рис. 1. Функциональная схема кристалла ПЦОС «Рената»

В основе принципа генерации лежит метод интегрирования в цифровой форме некоторой константы S до максимального значения результата интегрирования M . Для данного процессора предпочтительнее интегрировать от M до нуля (рис. 2).

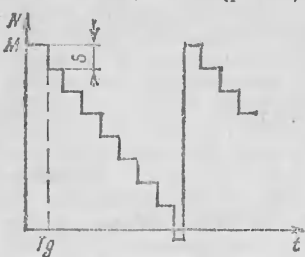


Рис. 2. Принцип генерации сигнала фиксированной частоты

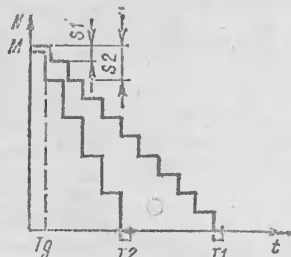


Рис. 3. Принцип изменения частоты сигнала

Период генерируемого сигнала (T_g) обратно пропорционален величине коэффициента S .

Постоянные S и M хранятся в специальной памяти процессора. В ПЦОС таких констант 16. Восемь положительных с адресами $KP0...KP7$ и восемь отрицательных с адресами $KM1...KM8$. Обозначения S и M определяют адреса регистров с соответствующими коэффициентами. Для удобства каждому из сорока обменных регистров ОЗУ, используемых в качестве регистров А и В, присваивается какое-либо наименование. Например, константа записана в регистре S, результаты генерации записаны в регистре OSC и т. д.

Программа генерации сигнала пилообразной формы

1. SUB OSC1, S ; отрицательное интегрирование
LDA DAR, OSC1 ; загрузка результата в DAR-регистр
ADD OSC1, KP4, LO1, CNDS ; проверка знака результата, если OSC < 0, то загрузка в OSC1 значения M-1
2. LDA OSC, OSC1 ; загрузка регистра результата
SUB OSC, KP4 ; образование двухполярного импульса
3. ABS OSC, OSC, LO1 ; нахождение модуля и умножение на 2
4. SUB OSC, KP4 ; образование двухполярного импульса
5. ADD OSC, OSC, LO1 ; умножение на 3 (минимизация 3-й гармоники)

Примечание. В регистре S записана константа, в регистре OSC — результат, Регистр OSC1 служит для хранения промежуточных результатов.

Профильтровав трапециевидный сигнал (п. 5 программы) фильтром низких частот, получим синусоидальный сигнал.

С помощью ПЦОС можно получить одновременно восемь сигналов различной частоты (т. е. лежащих в разных поддиапазонах) путем последовательного их вывода на каждый из восьми выходов. Для этого результаты помещаются в DAR-регистр и выполняется команда OUT. Синтезируемые частоты отличаются большой стабильностью, так как процессор тактируется внутренним кварцевым генератором.

Частоту генерируемого сигнала можно изменять внешним управляющим аналоговым или цифровым сигналом, а также по алгоритму, заданному программно. Процесс изменения частоты сводится к изменению коэффициента S при $M = \text{const}$ (рис. 3). Плавное изменение частоты генерации может осуществляться путем подачи управляющего напряжения на любой из аналоговых входов (например, IN0). Для переключения диапазона можно использовать, например, вход IN1, подавая «Log.0» или «Log.1».

Практически управление по частоте осуществляется так. Производится аналого-цифровое (А-Ц) преобразование входного управляющего сигнала. С помощью команды LDA DAR.KPO.ROO загружается «0» в DAR-регистр. Далее осуществляется опрос входа IN0. Время опроса должно быть достаточным для перезарядки конденсатора выборки и хранения и при тактовой частоте ПЦОС, равной 5 МГц, составлять шесть команд IN0 [2]. Знак и цифровое значение выборки определяются программно, начиная со старшего бита:

CVTS
NOP
CVT6
...
CVT0
NOP

Команды NOP необходимы для завершения переходных процессов при А-Ц преобразовании. По окончании А-Ц преобразования результат помещается в регистр S1.

Для получения переключения диапазона необходимо осуществить А-Ц преобразование по входу IN1 (LDA DAR.KPO.ROO.TN1). Так как вход IN1 используется в данном случае как цифровой, то достаточно определить только старший бит. Результат пересылается в регистр KON:LDA KON.DAR. Масштабирование в первом и втором диапазонах осуществляется командами LDA S.S1.R07 и LDA S.S1.R08.CND7 соответственно; $U_{\text{лх}}$ изменяется от 0 до 2 В. Если длина программы равна половине максимальной, т. е. 96, то при $f_{\text{кв}} = 5$ МГц период дискретизации равен 76,8 мкс [2], что соответствует частоте 13,02 кГц. Если задаться соотношением между максимальной частотой дискретизации и частотой сигнала равным 100, то верхняя граница диапазона (f_{max}) будет 130,2 Гц. В этом случае управляющий сигнал $U_{\text{вх}}$ должен быть равен максимальному значению. Нижняя граница диапазона частоты (f_{min}) будет соответствовать f_{max} (для преобразования с точностью 8 бит), т. е. $130,2/256 = 0,509$. При заданном соотношении частот число ступенек должно быть не менее 100. В этом случае для упрощения масштабирования код в регистре DAR, соответствующий $U_{\text{вх}}$, надо уменьшить в 128 раз (2^7), что осуществляется путем сдвига его на семь разрядов вправо.

Если необходимо получить меньшую частоту, производится дополнительный сдвиг вправо по выполнению условия, поступающего через вход IN1 в виде логического уровня.

Синусоидальный сигнал, генерируемый таким устройством, имеет большую стабильность частоты. Точность настройки определяется точностью А-Ц преобразования

(~0,4%). Следует отметить при этом амплитудную модуляцию пилообразного сигнала и, как следствие, — частотную модуляцию синусоидального сигнала. Это обусловлено применением команды ADD OSC1.KP4.LO1.CNDS. При выполнении команды LDA OSC1.KP4.LO1.CNDS такая модуляция отсутствует, а частота сигнала изменяется кратно T_g . Частотная модуляция выходного сигнала состоит в том, что некоторые периоды могут быть короче соседних на величину T_g . Частота появления таких периодов зависит от величины остатка $\Delta = M - nS$, где n — количество ступенек целое при $\Delta \leq S$ (число ступенек изменяется через период при $\Delta = S/2$).

Преимущества описанного генератора (по сравнению с синтезаторами частот, построенными по методу с табличной аппроксимацией или с накоплением фазы): широкие функциональные возможности (можно организовать управление по фазе и амплитуде, изменив лишь программу); высокая стабильность частоты (определяется кварцевым резонатором) при возможности плавной и практически безынерционной перестройки; высо-

кая надежность. Следует, однако, учитывать, что максимальная частота генерации сигнала заданной формы при фиксированной длине программы определяется быстродействием ПЦОС «Рената» и не превышает нескольких килогерц.

Адрес для справок: 290005, г. Львов, ул. Ватутина, д. 5, ЛПИ, т. 72-88-30.

ЛИТЕРАТУРА

- Хофф М. Е., Таунсенд М. Однокристалльный микрокомпьютер для обработки сигналов в реальном времени // Электроника. — 1979. — № 5. — С. 23.
- Марфевко К. С., Подлепецкий Б. И. БИС специализированных цифровых процессоров для обработки сигналов // Зарубежная электронная техника. — 1985. — № 3. — С. 3.
- Оппенгейм А., Шафер Р. Цифровая обработка сигналов. — М.: Связь, 1979.

Статья поступила 10 марта 1986 г.

УДК 681.3

А. М. Шапуло, Н. Н. Афанасьева

ОТЛАДОЧНЫЙ КОМПЛЕКС КОНТРОЛЛЕРА «ЭЛЕКТРОНИКА К1-20»

Комплекс предназначен для разработки, отладки и контроля функционирования систем управления на базе контроллера «Электроника К1-20» (новое обозначение — «Электроника МС 2702»).

Комплекс (рис. 1) состоит из контроллера с подключенным при необходимости пультом управления и микроЭВМ «Электроника ДЗ-28»*

с вводом данных во встроенном дополнительном ЗУ (ДЗУ); выводить информацию на цифровую индикацию и печать; запускать в работу требуемую программу или задачу; останавливать программу; инициировать адрес и данные в шаговом режиме; редактировать части и целую отлаживаемую программу; перемещать программу, исключая или вставляя слова.

Для работы комплекса необходимо

загрузить в ОЗУ микроЭВМ отлаживаемую программу контроллера и обслуживающую программу работы микроЭВМ.

Простейшая программа обслуживания содержит 30 шагов: вводится по шине «Ввод» (Вва, Ввв) первая половина информации магистрали адреса контроллера; затем — вторая; далее содержимое (находящееся в памяти микроЭВМ по адресу, равному сумме инверсии введенного адреса и базы) инвертируется и выводится на шину «Вывод» (X2, У2).

Исходное состояние сигналов ВВБОР, ВВВОД, СВЯЗЬ — «Лог.0», а ВВВ→МД, Т, ГОТОВНОСТЬ, СИП — «Лог.1».

При обращении контроллера к памяти по адресу $4000_{16} \dots 7999_{16}$ воз-

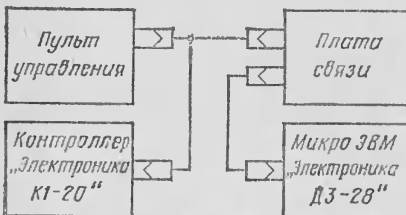


Рис. 1. Функциональная схема отладочного комплекса

ное обозначение — «15 ИПГ-32-003»), сопрягаемых между собой через плату связи. Плата связи (рис. 2) и микроЭВМ подключаются вместо ПЗУ контроллера, поэтому перемычки X12 и X13 контроллера, входящего в комплекс, должны быть разорваны.

Комплекс позволяет оперативно и многократно вводить необходимые программы работы контроллера и данные в память с контролем правильности ввода; записывать и хранить необходимые программы и мас-

* Теплицкий Ф. Н., Чуглаев Г. И., Фельд Б. А. Отладочный комплекс на основе ВУМС «Электроника ДЗ-28» // Электронная промышленность, — 1983. — 3. — С. 53.

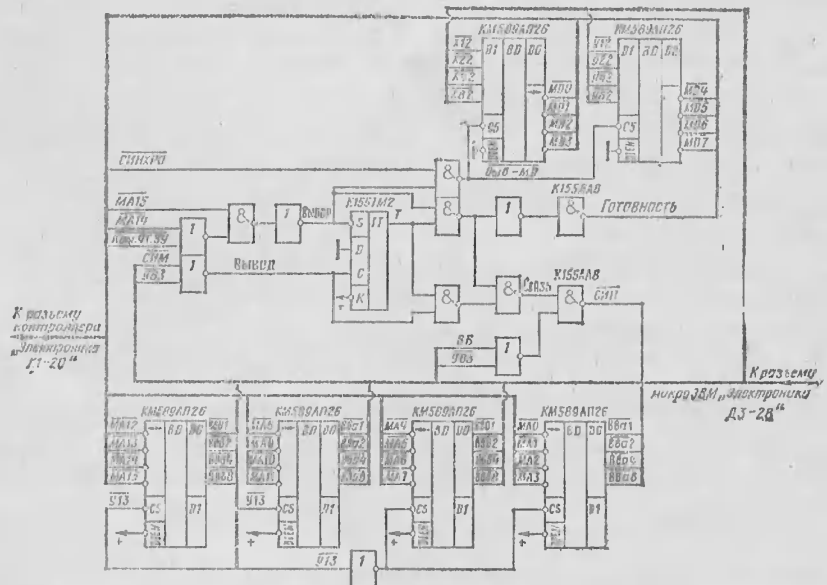


Рис. 2. Функциональная схема платы связи

никает сигнал ВЫБОР=«Лог.1», пропадает сигнал ГОТОВНОСТЬ.

Сигнал СВЯЗЬ устанавливается в «Лог.1», и становится возможным ввод адреса в микроЭВМ по программе обслуживания. При этом сигналы ВВ=«Лог.0» (ввод в микроЭВМ), У83=«Лог.0» (адрес платы связи), СИП становится равным «Лог.0» и первая часть адреса данных через первую половину буфера адреса с адресом У13=«Лог.0» вводится в микроЭВМ. После ввода сигналы ВВ и СИП восстанавливаются в «Лог.1».

Далее по обслуживающей программе вводится вторая часть адреса через вторую половину буфера адреса с адресом У13=«Лог.1». Последовательно сбрасываются сигналы ВВ и СИП, У83=«Лог.0», вводится данные,

и исходное состояние сигналов ВВ и СИП восстанавливается.

Затем данные выводятся из микроЭВМ. По сигналам СИМ=«Лог.0» и У83=«Лог.0» сигнал ВЫВОД устанавливается равным «Лог.1». Триггер сбрасывается (Т=«Лог.0»), запрещается сигнал СИП (СВЯЗЬ=«Лог.0»). Возникает сигнал ГОТОВНОСТЬ и становится возможным ввод данных из микроЭВМ в контроллер через буфер, открытый после сигнала СИХРО при сигнале ВЫБОР=«Лог.1».

Контроллер, приняв данные, снимает сигнал ВЫБОР («Лог.0»). Триггер устанавливается (Т=«Лог.1»). Сигнал связи (СВЯЗЬ=«Лог.1») разрешает передачу сигнала СИП. При этом если сигнал ВВ=«Лог.0», то сигнал СИП=«Лог.0». Затем последователь-

но устанавливаются в «Лог.1» сигналы ВВ, СИМ и СИП. Сигнал ВЫВОД сбрасывается в «Лог.0». Цикл программы обслуживания закончен, можно начинать новый ввод-вывод.

Контроллер может снова обращаться к памяти микроЭВМ сразу после предыдущего обращения. В этом случае предыдущий вывод микроЭВМ продолжает заканчиваться, так как триггер сбрасывается (Т=«Лог.0») только в начале нового вывода данных. Изменяются только сигналы ВЫБОР, ВВ→МД и ГОТОВНОСТЬ, и контроллер ждет появления сигнала ГОТОВНОСТЬ.

Адрес для справок: 734025, Душанбе, ул. Горького, 26, кв. 1, Шапуло А. М.

Статья поступила 20 октября 1986 г.

УДК 681.322.1

И. Л. Мышкин, А. С. Щербаков

ОДНОПЛАТНЫЙ МИКРОКОНТРОЛЛЕР «ПРОЕКТ-80»

Микроконтроллер предназначен для работы в системах управления и автоматизации, средствах проектирования устройств на базе МПК серии КР580, в составе персонального компьютера. Конструктивное исполнение контроллера позволяет устанавливать его в диалоговый вычислительный комплекс (ДВК).

В состав аппаратных средств контроллера входят центральный процессор на базе МП КР580ВМ80А, по-

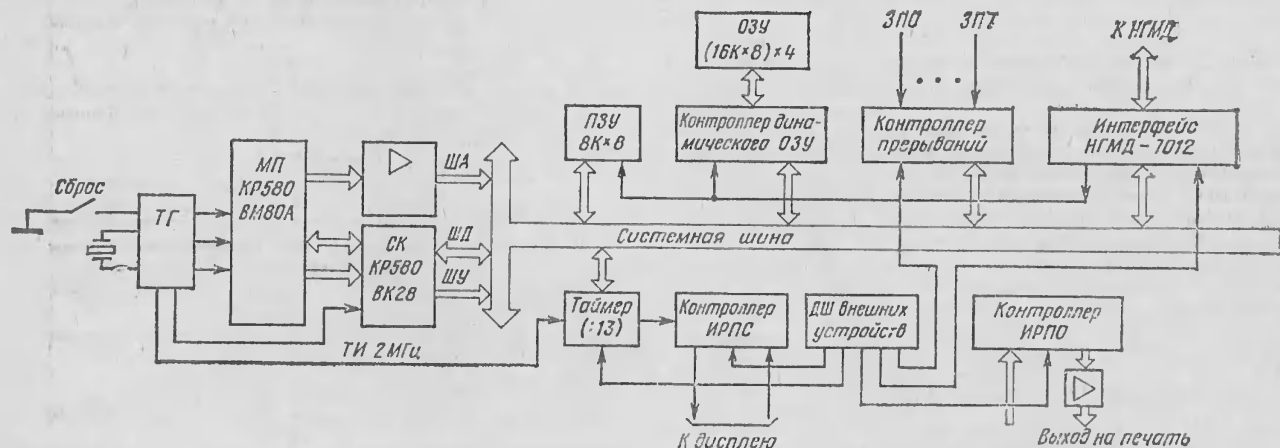
стоянная память с системным монитором и загрузчиком общим объемом до 8К байт на микросхемах К573РФ2, динамическое ОЗУ с «прозрачным» управлением общим объемом 64К байт, контроллер прерываний на восемь каналов, интерфейс накопителя на гибких магнитных дисках ГМД-7012, интерфейс радиальный последовательный (ИРПП) (см. рисунок).

Программное обеспечение контроллера включает управляющий мони-

тор и программу загрузки операционной системы ОС-1800 (программно совместима с ОС СР/М), которая загружается с гибких дисков ГМД-7012.

Сравнительный анализ эффективности работы микроконтроллера «Проект-80» и микроЭВМ СМ-1800 в операционной среде ОС-1800 показал, что при работе программ длиной более 30К байт и частом обращении к ОЗУ (сортировка, компоновка) время их выполнения на микроконтроллере в среднем на 35..40% меньше, чем на СМ-1800.

Сообщение поступило 10 ноября 1986 г.



Структурная схема микроконтроллера «Проект-80»

ОРГАНИЗАЦИЯ ДИНАМИЧЕСКОЙ ПАМЯТИ МИКРОСИСТЕМ

(Продолжение цикла. Начало см. АЭЛЭ 5, 6, 1986 г., АЭЛЭ 1—3, 1987 г.)

Рассмотренные в работе устройства памяти динамического типа дают возможность расширить ресурсы одноплатных микроконтроллеров семейства ММС1200 и ММС1210. На их основе спроектирован новый однокристалльный 8-разрядный микроконтроллер ММС1206 с ОЗУ объемом до 64К байт и 16-разрядный модуль памяти комбинированного типа ММС3702, имеющий 128К байт ОЗУ и до 128К байт ПЗУ. Устройства стыкуются с одноплатными МК семейства ММС, расширяя их ресурсы, и могут быть использованы для проектирования различных систем памяти.

Динамические ОЗУ серии К565 являются основной элементной базой для разработчиков микросистем при построении систем памяти большой емкости. Приборы памяти динамического типа потребляют меньше энергии и характеризуются более высокой плотностью упаковки, которая позволяет в 16-контактном корпусе с вертикальными выводами разместить кристалл емкостью до 256К байт.

Вместе с этим интерфейс приборов динамической памяти более сложен и для их нормального функционирования необходимо: мультиплексирование адресов, формирование специальных стробов приема адреса и управления работой прибора, регенерация состояния прибора, обеспечивающая сохранность информации, арбитраж с целью разрешения конфликтных ситуаций одновременного поступления нескольких запросов. Кроме того, динамические ОЗУ не всегда обеспечивают передачу данных со скоростью быстродействующих процессоров, поэтому требуется формирование ряда периодов ожидания. Эти операции выполняются с помощью специальной интерфейсной и управляющей логики. Прием адреса в динамических ОЗУ производится в два этапа: адрес строки (запоминается в ОЗУ по спаду управляющего сигнала RAS (вывод 4) и адрес столбца (по спаду сигнала CAS (вывод 15)).

Существует три основных цикла обращения к памяти [1]: запись, считывание и регенерация (рис. 1).

При проектировании конкретного контроллера динамического ОЗУ должна учитываться скорость выборки динамического ОЗУ, характеризуемая двумя параметрами: време-

нем выборки, отсчитываемым от момента перехода в активное состояние сигнала RAS, и временем выборки, отсчитываемым от активации сигнала CAS.

Особенность динамических приборов памяти — периодическая регенерация содержимого для продолжительного хранения информации. Состояние «Log.0» или «Log.1» ячейки памяти определяется наличием заряда на запоминающей емкости. Однако если к элементу долгое время не обращаться, то за счет тока утечки данные могут быть утеряны. Для предотвращения этого эффекта необходимо через 2 мс считывать информацию из каждой ячейки памяти. При этом усилитель считывания автоматически восстанавливает заряд на запоминающей емкости, обеспечивая сохранность данных на некоторое время.

Так как в динамическом ОЗУ каждый столбец имеет свой усилитель считывания, то процедуру регенерации можно выполнять одновременно для всей строки. Микросхемы К565РУ5 и К565РУ7 имеют 128 и 256 строк соответственно. Регенерация выполняется в укороченном цикле — передача только адреса строки, подлежащей регенерации (внешних операций с данными в этом цикле нет).

Следует отметить, что регенерация данных адресуемой строки выполняется автоматически в любом цикле обращения к памяти, что может быть использовано для сохранения инфор-

мации ОЗУ. Однако в общем случае нельзя гарантировать периодическое обращение ко всему диапазону строк прибора памяти каждые 2 мс. Возникает необходимость в разработке специальных схем принудительной регенерации.

Существует несколько стандартных алгоритмов выполнения процедуры регенерации:

алгоритм пакетной регенерации подразумевает выполнение регенерации всей памяти непрерывным блоком каждые 2 мс. Между процедурами память свободна для выполнения своих прямых функций. Такой способ может существенно удлинить время реакции микросистемы на любое внешнее событие.

Если данные каждой строки регенерируют каждые 2 мс, то для сохранения информации несущественно, как и в какой последовательности отдельные циклы распределены по всему интервалу регенерации. Наиболее часто используют равномерное распределение отдельных циклов. Если МП обращается к памяти в момент выполнения очередного цикла регенерации, контроллер приостанавливает обращение до завершения текущего цикла. При этом время выполнения операции записи или считывания данных может возрасти на время цикла регенерации, а увеличение времени отклика на внешнее событие оказывается в прямой зависимости от длины процедуры обслуживания этого события.

Третий алгоритм называется «прозрачной» регенерацией, так как не заметен для микропроцессора и не влияет на скорость выполнения программ. В ряде микропроцессоров, например КР580ВМ80, 8085 и др., между двумя любыми циклами обращения к памяти существует интервал времени, достаточный для выполнения одного цик-

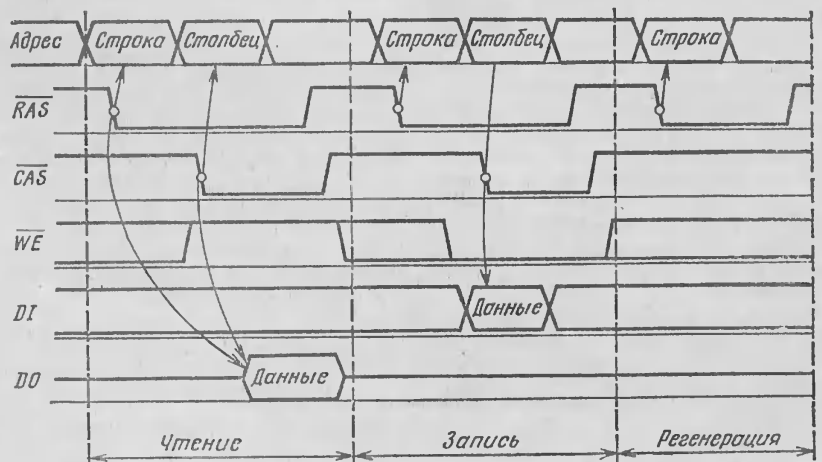


Рис. 1. Динамические ОЗУ серии К565, временные диаграммы обращения к памяти

ла регенерации ОЗУ. Если регенерация привязать к окончанию циклов записи-чтения и проводить в этих окнах, то она может быть выполнена незаметно для микропроцессора.

В режимах ожидания готовности внешних устройств WAIT или остановки HALT МП долгое время не производится обращения к памяти, поэтому должна быть предусмотрена возможность автоматического продолжения циклов регенерации в периодическом режиме при отсутствии циклов обращения к памяти. Устройства памяти, построенные таким образом, называются квазистатическими. Для разрешения конфликтной ситуации, возникающей при обращении к памяти во время регенерации ОЗУ, вводится специальная логика, входящая в состав контроллера ОЗУ. Такой метод организации памяти позволяет освободить микропроцессор и магистраль МС, однако приводит к некоторому усложнению аппаратуры контроллера.

В быстродействующих МП отсутствуют временные окна, достаточные для выполнения циклов регенерации ОЗУ, поэтому использование «прозрачного» алгоритма регенерации нецелесообразно.

Главная часть памяти — накопитель. Простейшие системы ОЗУ статического типа включают только накопитель, а динамические состоят также из мультиплексора адресов, контроллера памяти, интерфейсной схемы.

Память микросистем с 8-разрядной шиной данных на базе микропроцессоров КР580ВМ80, 8085, 8088, 80188 организована в виде линейного блока. Каждый байт накопителя адресуется независимо с помощью системной адресной шины АDR 15(ADR19)...ADR0 (за один цикл обращения к памяти реализуется обмен одним байтом).

Слова в памяти занимают два соседних байта. В байте с младшим адресом хранится младшая часть слова. Его адрес служит адресом слова и может быть как четным, так и нечетным. Двойные слова размещаются в памяти в виде двух соседних слов. Младшее слово располагается по младшему адресу, а его адрес является адресом двойного слова. Слова, а также двойные слова считываются или записываются за два и четыре цикла соответственно, начиная с младших байтов.

Память 16-разрядных МС на основе микропроцессоров К1810ВМ86, 80186 и 80286 организована линейным набором 16-разрядных ячеек, разбитых на два байтовых банка: Н-банк (НВ) и L-банк (ЛВ) по 512К байт для К1810ВМ86 и 80186, 8М байт для 80286. Банк НВ связан со старшим байтом 16-разрядной шины данных, ЛВ состыкован с младшей половиной шины данных (рис. 2). Хра-

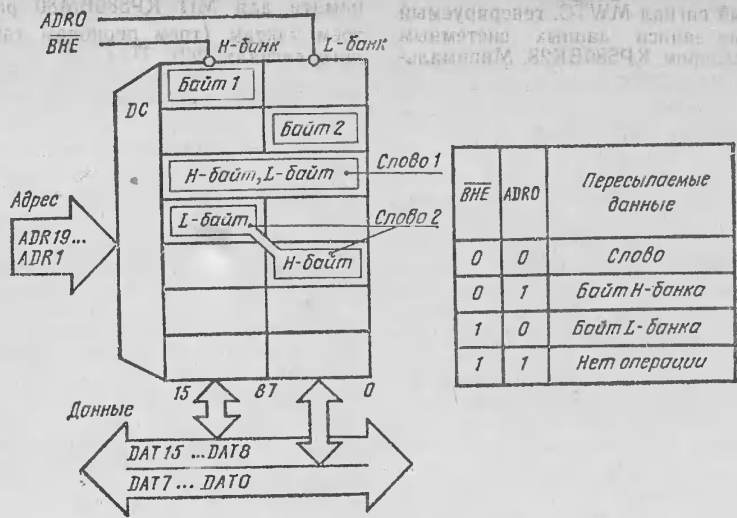


Рис. 2. Организация физической памяти

нение байтов, слов и двойных слов в памяти МС аналогично случаю 8-разрядной системной магистрали.

Для адресации 16-разрядных ячеек физической памяти используются адресные линии АDR19(ADR23)...ADR1 системной магистрали. Линия АDR0 совместно с дополнительным сигналом $\overline{ВНЕ}$ (Byte High Enable) реализует функцию независимой выборки банков ЛВ и НВ. Если доступ к L-банку разрешается только при $АDR0=0$, то доступ к Н-банку открыт, когда $\overline{ВНЕ}=0$. При этом информация передается через старший байт шины данных. Сигнал $\overline{ВНЕ}$ служит признаком того, что старшая половина шины данных АDR15...DAT8 участвует в обмене. При $АDR0=\overline{ВНЕ}=0$ выполняется обмен словом. Записывание байтов, а также размещение по четному адресу слов выполняется за один цикл обращения к памяти. Доступ к слову с нечетным адресом реализуется за два цикла: по старшей половине шины данных передается младший байт слова ($АDR0=1, \overline{ВНЕ}=0$), по младшей половине — старший. Учитывая это обстоятельство, слова в памяти рекомендуется располагать по четным адресам, что особенно важно при организации системного стека, операции с которым всегда 16-разрядные.

При использовании 16-разрядных систем памяти с 8-разрядными шинами данных необходимо:

$$\left. \begin{aligned} \overline{ВНЕ} &= АDR0 = АDR0^*, \\ DAT_{n+8} &= DAT_n = DAT_n^*, \quad n = 0, 1, \dots, 7 \end{aligned} \right\} (1)$$

Сигналы 8-разрядной магистрали отмечены знаком *.

При отображении младшей половины адресов 8-разрядной МС в L-банк, а старшей — в Н-банк, используется подключение:

$$\left. \begin{aligned} \overline{ВНЕ} &= АDR0 = АDRN^* \\ АDR_{n+1} &= АDR_n^*, \quad n = 0, 1, \dots, N-1 \\ DAT_{n+8} &= DAT_n = DAT_n^*, \quad n = 0, 1, \dots, 7 \end{aligned} \right\} (2)$$

где N определяется шириной адресной шины.

Для построения накопителя динамического ОЗУ на базе серии К565 используются блок из 8 или 16 микросхем, по одному прибору на разряд ячейки памяти. При этом все адресные шины и линии управления микросхем ОЗУ соединяются между собой, образуя адресную шину и набор линий управления блоком. При построении больших систем памяти используют несколько блоков, выбор которых определяется сигналами RAS и CAS.

Разработан ряд стандартных однокристалльных контроллеров динамической памяти [2]. Их использование желательно, если технические данные на микросхемы удовлетворяют требованиям на организацию памяти. В противном случае задача проектирования контроллера должна быть выполнена разработчиком.

Рассмотрим устройство оперативной памяти емкостью 64К байта с «прозрачной» регенерацией, встроенное в одноплатный микроконтроллер МС1206. Данный МК является модификацией 8-разрядного модуля МС1204 [3], выполненной с целью расширения резидентной части ОЗУ до 64К байт.

На рис. 3 приведены диаграммы чтения-записи данных для ЦП на базе микропроцессора КР580ВМ80, системного контроллера КР580ВК38 и генератора тактовых импульсов КР580ГФ24. Пунктиром обозначен си-

стемный сигнал MWTC, генерируемый в цикле записи данных системным контроллером КР580ВК28. Минимальный цикл между обращениями к

памяти для МП КР580ВМ80 равен трем тактам (трем периодам тактового сигнала Ф2). При номинальной частоте кварцевого резонатора

$f_{ХТАЛ} = 1/T = 18$ МГц длительность такта составляет 0,5 мкс. Для реализации «прозрачной» регенерации памяти имеется достаточно времени даже при использовании микросхем К565РУ5Д с временем цикла 450 нс. Длительности сигнала чтения MRDC и записи MWTC также превышают значение 0,5 мкс, следовательно, можно организовать обмен данными в синхронном режиме без использования сигнала READY микропроцессора.

Схема модуля памяти объемом 64К байт, работающего без тактов ожидания и использующего «прозрачный» алгоритм регенерации, представлена на рис. 4. Полный цикл регенерации ОЗУ осуществляется по крайней мере за 200 мкс, что делает возможным использование микросхем К565РУ5Д с периодом регенерации 1 мс.

Устройство памяти содержит: накопитель, выполненный на схемах DD1...DD8 (К565РУ5Д); мультиплексор адреса, реализованный на программируемом ПЗУ DD11, DD12, DD13 (К565РТ5); счетчик адреса ре-

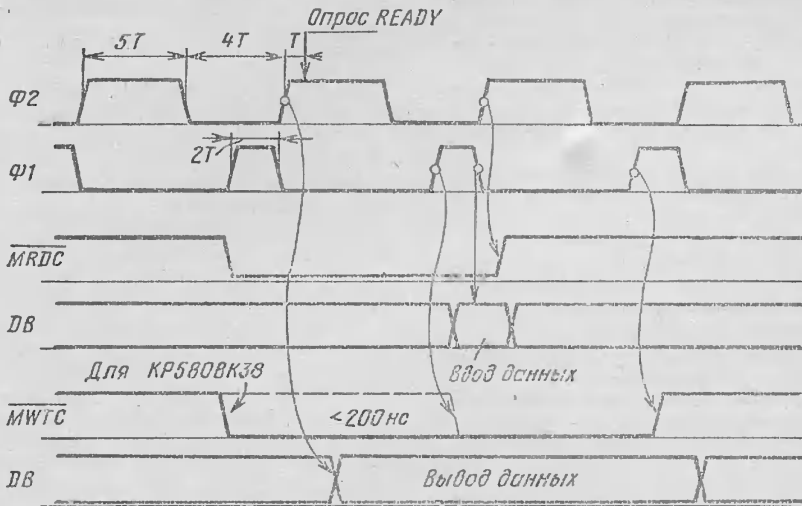


Рис. 3. Временные диаграммы обращения к памяти ЦП на базе КР580ВМ80

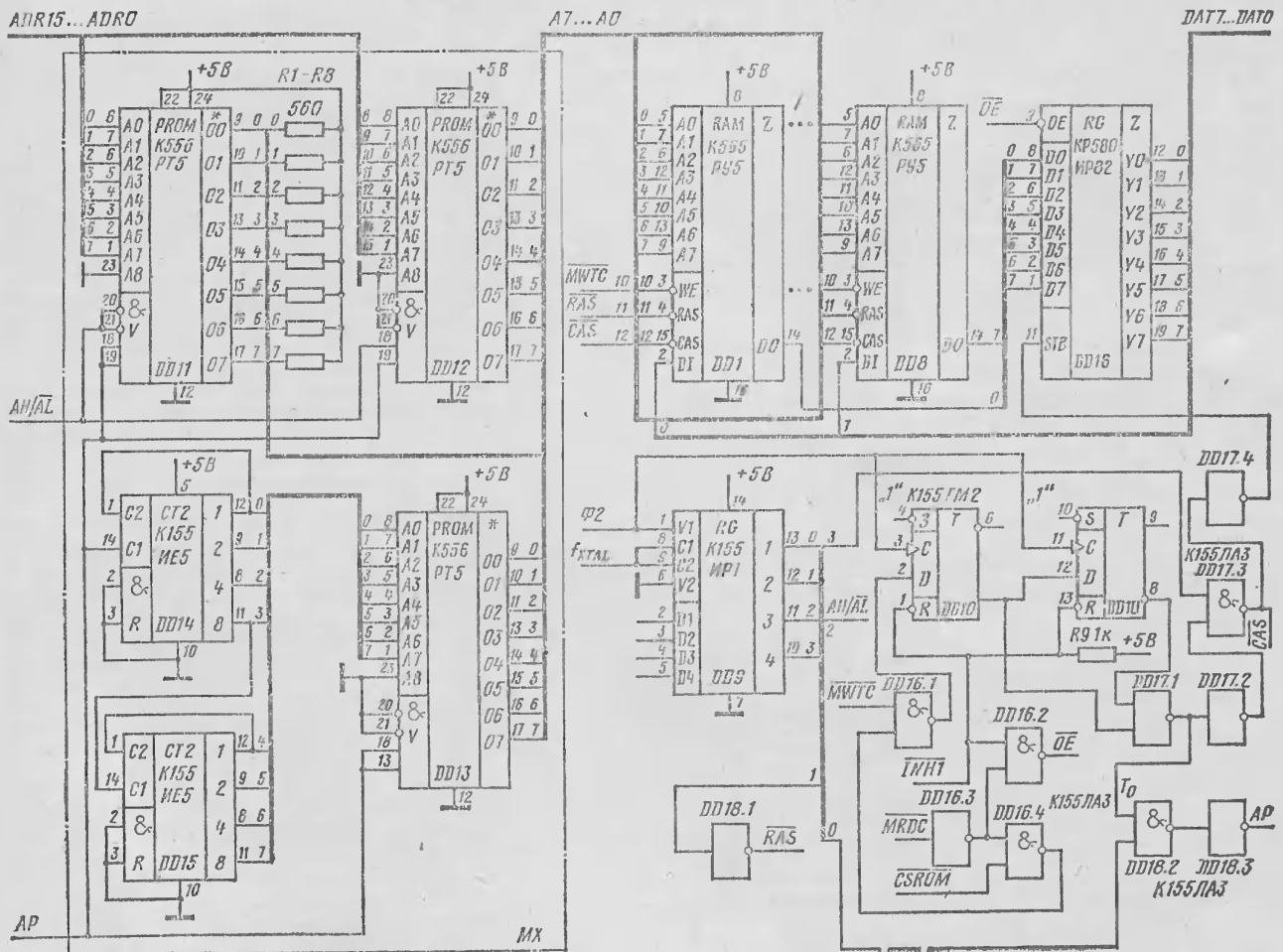


Рис. 4. Устройство памяти 64К байт

генерации DD14 и DD15 (K155HE5); схему управления, включающую регистр DD9 (K155IP1), два триггера DD10 и три стандартные микросхемы малой логики DD16, DD17, DD18 (K155ЛА3) (рис. 5).

При поступлении сигнала \overline{MRDC} или MWTC по фронту Ф2 формируется временной интервал T_0 обращения к ОЗУ. Временная последовательность подачи управляющих сигналов RAS и \overline{CAS} , сигнала подключения адресов строк и столбцов AH/AL, а также сигнала подключения адреса регенерации AP задается с помощью регистра сдвига DD9. После завершения обмена устройство переходит в режим регенерации с одним сигналом RAS и остается в этом режиме до следующего обращения к ОЗУ.

Рассмотренное устройство памяти 64К байт с «прозрачной» для МП регенерацией ориентировано на работу с конкретным микропроцессором КР580ВМ80. В микропроцессорах К1810ВМ86, К1810ВМ88 используется более высокая тактовая частота и предварительная выборка командной последовательности глубиной до 6 байт. Поэтому фаза испол-

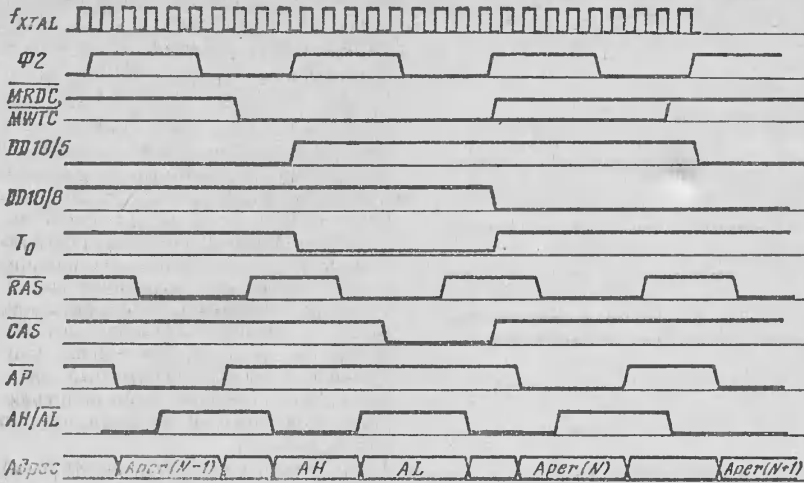


Рис. 5. Временные диаграммы работы схемы управления динамическим ОЗУ

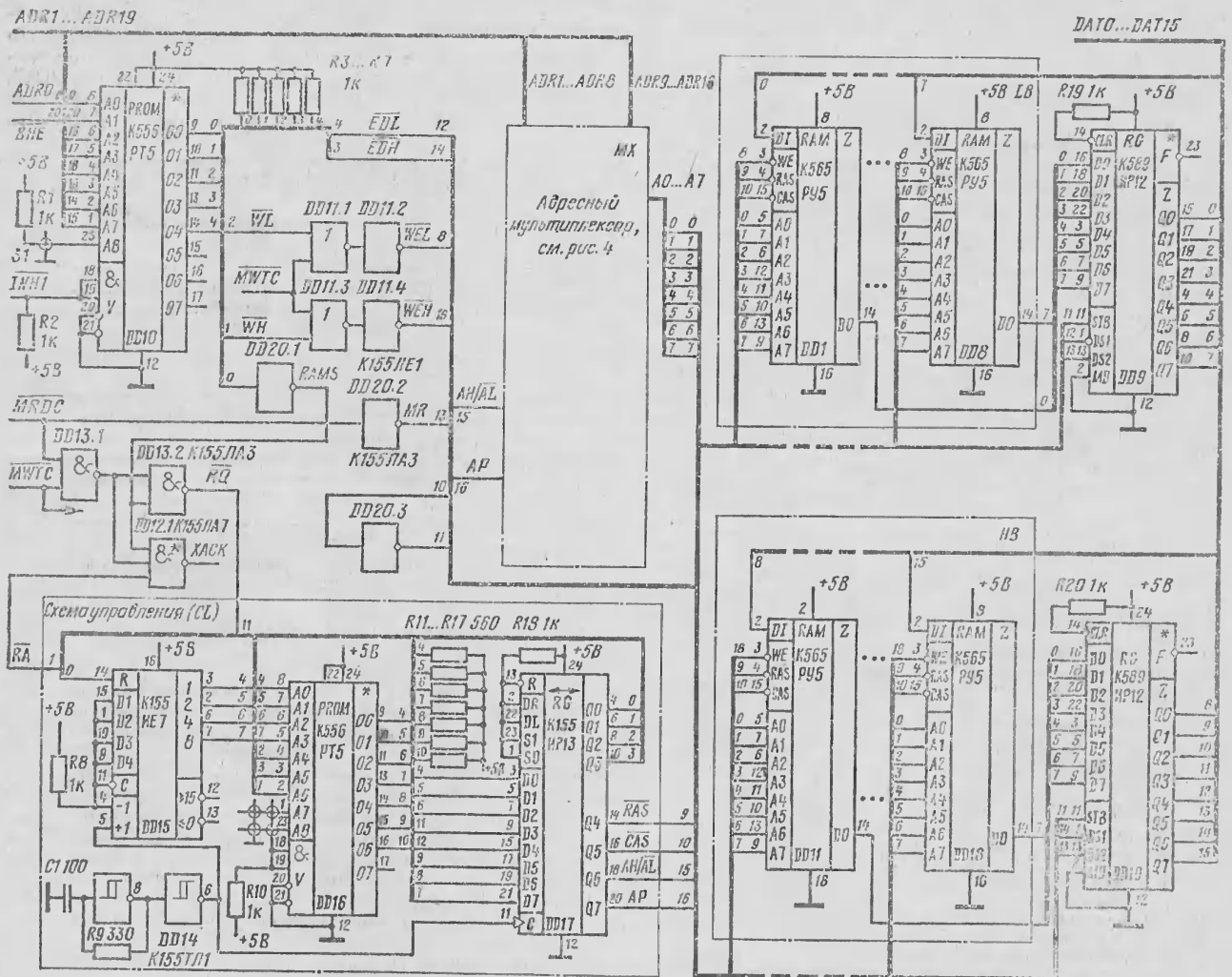


Рис. 6. Устройство памяти 128К байт

нения команды оторвана от фазы ее выборки, а между отдельными циклами обращения к ОЗУ отсутствуют временные интервалы, достаточные для очередной регенерации строки. Практическая возможность построения квазистатического ОЗУ для таких микропроцессоров сильно ограничена или отсутствует совсем.

Организация устройства памяти объемом 128К байт ориентирована на 16-разрядный интерфейс типа И41 (рис. 6). При непосредственном подключении к системной магистрали одноплатного микроконтроллера ММС1212 [4] его ОЗУ расширяется до 130К байт. Данное устройство было использовано для построения модуля памяти комбинированного типа ММС3702, в состав которого кроме 128К байт ОЗУ входит до 128К байт ПЗУ.

Интерфейсная логика осуществляет сопряжение модуля с системной шиной типа И41 с точностью до инверсии линий адреса и данных. Логика содержит дешифратор адреса DD10, формирователь ответного сигнала подтверждения обмена ХАСК, а также выходные буферные регистры DD9, DD19.

Реализованный на ПЗУ К556РТ5 (DD10) дешифратор адреса кроме сигнала выборки устройства RAMS генерирует сигналы разрешения записи данных, поступающие после логического совпадения с MWTC на входы WE приборов памяти, в старший и младший банки памяти WH и WL соответственно, а также сигналы разрешения выходных буферов старшего байта слова EDH и EDL, поступающие на входы DS1 микросхем К589ИР12. Состояние входа S1 дешифратора адреса задает режим работы устройства. При S1=0 реализуется 16-разрядный режим работы с системной магистралью MC на базе К1810ВМ86. В этом случае

$$\left. \begin{aligned} WH &= BHE \\ WL &= ADRO \\ EDL &= EDH = RAMS \end{aligned} \right\} (3)$$

При S1=1 устройство работает с 8-разрядной системной шиной, формируемой ЦП на базе КР580ВМ80. Логика работы дешифратора соответствует выражениям, удовлетворяющим (1):

$$\left. \begin{aligned} WH &= ADRO \\ WL &= ADRO \\ EDH &= RAMS \cdot ADRC \\ EDL &= RAMS \cdot ADRO \end{aligned} \right\} (4)$$

Соединение старшей и младшей частей шины данных должно быть выполнено на плате ЦП.

Формирование сигнала S1, переключающего режим работы устройства, можно производить автоматически в момент начального пуска или перезапуска системы (рис. 7). Логический стартовый адрес микропро-

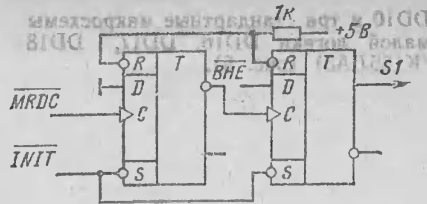


Рис. 7. Схема автоматической установки режима работы

цессоров К1810ВМ86 и 80186 (0F1F0H:0000H), а также микропроцессора 80286 (0F000H:0F1F0H) всегда четен, поэтому первый цикл обращения к памяти со стороны МП с 16-разрядной внешней шиной данных состоит в чтении слова, при котором BHE=0. В микросистемах с 8-разрядным интерфейсом построенных на базе КР580ВМ80, 8085, К1810ВМ86 или 80186 все циклы обращения к памяти являются шестнадцатичными байтовыми. При этом первый цикл чтения памяти будет выполняться при BHE=1.

Работа модуля логик возможна только в случае, если системный сигнал запрета ОЗУ INH деактивирован. Этот сигнал формируется внешними модулями по схеме «мониторинг ИЛИ» и дает возможность реа-

лизовать программно-управляемое наложение различных специальных буферов и сегментов ПЗУ на системное ОЗУ. При INH1-RAMS=1 строб MRDC или MWTC фиксирует запрос обмена с памятью RQ и сбрасывает флаг ХАСК, который переводит ЦП в режим ожидания WAIT.

Схема управления генерирует сигнал RA, который приводит к активности флага ХАСК и завершению обмена, определяет алгоритм работы устройства памяти: осуществляет арбитраж, управляет регенерацией и адресным мультиплексором, формирует управляющие сигналы для приборов памяти динамического типа. Схема представляет собой универсальный микропрограммный автомат работа которого легко подстраивается под требуемую организацию памяти (рис. 8).

При отсутствии запросов (RQ=1) реализуется периодическая регенерация ОЗУ (рис. 8, а). Период определяется емкостью счетчика DD15 и частотой генератора DD14. Устройство работает при $f \leq 10$ МГц, что определяется задержкой распространения сигналов в DD15 и DD17. В конкретном случае период $T = 1,6$ мкс, что позволяет увеличить

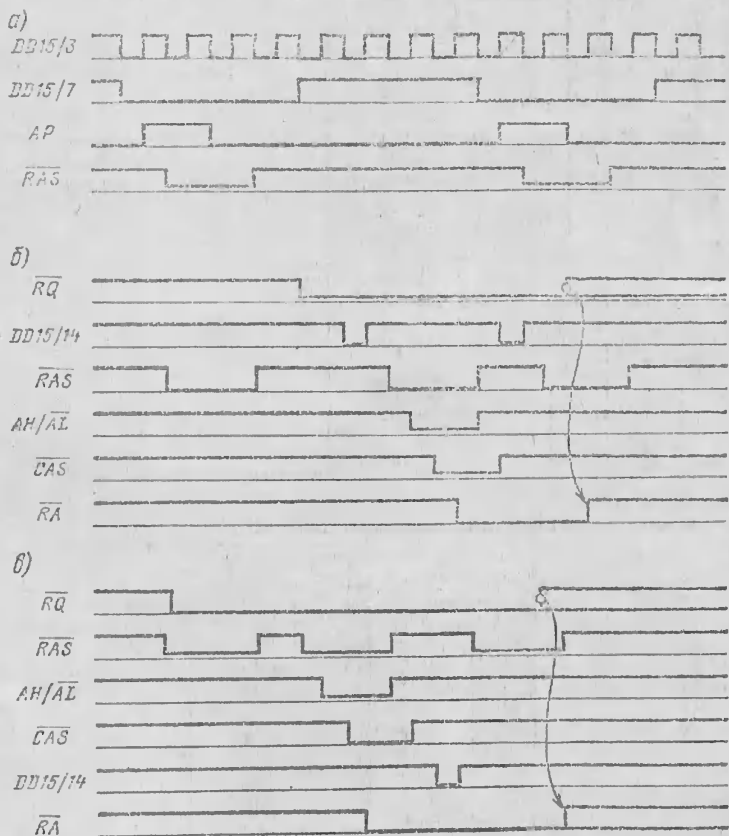


Рис. 8. Временные диаграммы работы схемы управления: а — регенерация; б — запрос обмена в режиме ожидания, в — запрос обмена в режиме регенерации

Рис. 9. Карта прожига ПЗУ DD11... DD13

Рис. 10. Карта прожига ПЗУ DD16

Рис. 11. Карта прожига ПЗУ DD10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.000.	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
.010.	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
.020.	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
.030.	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
.040.	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
.050.	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
.060.	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
.070.	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
.080.	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
.090.	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
.0A0.	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D
.0B0.	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E
.0C0.	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F
.0D0.	0D	0E	0F	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F	0F
.0E0.	0E	0F	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F	0F	0F
.0F0.	0F	0A	0B	0C	0D	0E	0F	0A	0B	0C	0D	0E	0F	0F	0F	0F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.000.	F0	F8	B8	B8	B0	F0	F0	F0	F0	F0	F0	F0	F0	F0	F0	F0
.010.	F6	FE	BE	BE	B6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6
.020.	F4	FC	BC	BC	B4	F4	F4	F4	F4	F4	F4	F4	F4	F4	F4	F4
.030.	F6	FE	BE	BE	B6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6
.040.	F2	B2	A2	B2	B2	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1
.050.	F6	FE	BE	BE	B6	F3	F3	F3	F3	F3	F3	F3	F3	F3	F3	F3
.060.	F6	FE	BE	BE	B6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6
.070.	F6	FE	BE	BE	B6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6	F6
.080.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.090.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0A0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0B0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0C0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0D0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0E0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0F0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.000.	E0	E4	E2	E6	E0	E4	E2	E6	FF	FF	FF	FF	FF	FF	FF	FF
.010.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.020.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.030.	FF	FF	FF	FF	FF	FF	FF	C7	C7	C7	C7	FF	FF	FF	FF	FF
.040.	E0	E4	E2	E6	E0	E4	E2	E6	FF	FF	FF	FF	FF	FF	FF	FF
.050.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.060.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.070.	FF	FF	FF	FF	FF	FF	FF	FF	C7	C7	C7	FF	FF	FF	FF	FF
.080.	E0	E4	E2	E6	E0	E4	E2	E6	FF	FF	FF	FF	FF	FF	FF	FF
.090.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0A0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0B0.	FF	FF	FF	FF	FF	FF	FF	FF	C7	C7	C7	FF	FF	FF	FF	FF
.0C0.	E0	E4	E2	E6	E0	E4	E2	E6	FF	FF	FF	FF	FF	FF	FF	FF
.0D0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0E0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0F0.	FF	FF	FF	FF	FF	FF	FF	FF	C7	C7	C7	FF	FF	FF	FF	FF

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.100.	EA	F4	EA	F4	EA	F4	EA	F4	FF	FF	FF	FF	FF	FF	FF	FF
.110.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.120.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.130.	FF	FF	FF	FF	FF	FF	FF	FF	CF	DF	CF	DF	FF	FF	FF	FF
.140.	EA	F4	EA	F4	EA	F4	EA	F4	FF	FF	FF	FF	FF	FF	FF	FF
.150.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.160.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.170.	FF	FF	FF	FF	FF	FF	FF	FF	CF	D7	CF	D7	FF	FF	FF	FF
.180.	EA	F4	EA	F4	EA	F4	EA	F4	FF	FF	FF	FF	FF	FF	FF	FF
.190.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.1A0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.1B0.	FF	FF	FF	FF	FF	FF	FF	FF	CF	D7	CF	D7	FF	FF	FF	FF
.1C0.	EA	F4	EA	F4	EA	F4	EA	F4	FF	FF	FF	FF	FF	FF	FF	FF
.1D0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.1E0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.1F0.	FF	FF	FF	FF	FF	FF	FF	FF	CF	D7	CF	D7	FF	FF	FF	FF

необходимые 256 циклов регенерации за время около 400 мкс.

При поступлении запроса RQ возможны две ситуации: автомат находится в режиме ожидания (состояние счетчика DD15 > 4) (рис. 8, б) и в режиме регенерации (рис. 8, в). В первых двух микротактах проводится анализ длительности сигнала RQ. При положительном исходе осуществляется сброс счетчика DD15 и производится цикл обмена с динамическим ОЗУ. Сигнал подтверждения обмена RA появляется через один микротакт после начала импульса CAS. Сразу же после окончания обмена схема управления переходит к очередному циклу регенерации. Таким образом осуществляется подсинхронизация фазы регенерации к очередному циклу обращения. В случае, если запрос RQ пришел во время регенерации, то выполнение обмена будет задержано до окончания цикла регенерации.

Наличие неиспользованных входов A7 и A8 ПЗУ DD16 позволяет реализовать несколько алгоритмов автомата, например, для интерфейса с прямым или инверсным сигналом ХАСК. Можно увеличить период регенерации, добавив еще несколько триггеров. В некоторых случаях полезно ввести «полупрозрачный» режим работы, когда сигнал ХАСК формируется только при попадании очередного запроса на цикл регенерации.

Карты прожига микросхем ПЗУ приведены на рис. 9, 10, 11. Предполагается, что оперативная память располагается в диапазоне 000000H...01FFFFH адресов. При этом выход Q5 прибора DD10 использовался для выбора ПЗУ, лежащего в диапазоне верхних адресов 0E0000H...0EFFFFH и размещаемого на одной плате с ОЗУ.

Справки по телефону: 408-62-44, Москва

ЛИТЕРАТУРА

1. Микросхема динамического ОЗУ K565PY5 / Справочная информация // Микропроцессорные средства и системы.—1985.— № 3.
2. Microsystem Components Handbook (Order 230843-002).— Intel Corp., USA. Vol. 2.—1985.
3. Щелкунов Н. Н., Дианов А. П. Универсальный одноплатный микроконтроллер // Микропроцессорные средства и системы.—1986.— № 5.— С. 65—69.
4. Щелкунов Н. Н., Дианов А. П. Одноплатный 16-разрядный микроконтроллер общего назначения // Микропроцессорные средства и системы.—1987.— № 1.— С. 77.

Статья поступила 27 августа 1987 г.

ОБМЕН ОПЫТОМ ПО ЭКСПЛУАТАЦИИ БЫТОВОЙ ПЭВМ „ЭЛЕКТРОНИКА БК-0010“

УДК 681.3.06

В. Т. Монахов

ПРОГРАММЫ ЛЮБИТЕЛЕЙ ДЛЯ БЫТОВОЙ ПЕРСОНАЛЬНОЙ ЭВМ «ЭЛЕКТРОНИКА БК-0010»

Бытовой компьютер «Электроника БК-0010» благодаря организации продажи через торговую сеть магазинов-салонов «Электроника» стал одним из наиболее распространенных в стране.

В Москве многие обладатели этих компьютеров объединились в секцию при клубе компьютерного всеобуча «Интерфейс». Секция имеет в своем распоряжении более 200 программ, созданных любителями-энтузиастами и являющихся собственностью автора или авторского коллектива (в комплект поставки БК-0010 входит всего пять программ).

Среди 200 программ есть разные по качеству, есть слабо документированные, но большинство программ успешно используется и вполне подготовлено к тиражированию.

По нашему мнению, после соответствующей проверки и отбора наиболее ценных программ по согласованию с авторами можно было бы пополнить комплект поставки БК-0010, поступающий в торговлю. Все программы хранятся на магнитной ленте и вводятся в ЭВМ с кассетного магнитофона.

Системные программы. Среди них интерпретатор языка Бейсик микроЭВМ «Электроника 60», адаптированный для БК-0010. Программа занимает около 7К байт в ОЗУ. Под программы пользователя остается 9...17К байт. Еще один интерпретатор Бейсика (называемый «Вильнюсским») отличается более высокой скоростью работы, значительным расширением по сравнению с Бейсиком «Электроника 60», но занимает объем более 9К байт, что несколько снижает возможности пользователей.

Коллективом энтузиастов из г. Рига во главе с Ю. Я. Кузьминым разработано большое число полезных программ и среди них адаптация известной системы программирования учебных программ РИГА с Т-языком, располагающая богатыми выразительными возможностями (речевой вывод информации, текстовый экраный редактор, специальный алфавит, возможность простого программирования динамических графических программ и т. п.). Использованный в системе РИГА Т-язык прост в изучении, хотя и весьма своеобразен. Проходит стадию тестирования у пользователей компилятор с языка ассемблера, позволяющий создавать программы непосредственно в машинных командах.

Для разработки программ в машинных командах предназначен микроотладчик и его усовершенствованные модификации: АСМ и Отладчик-3. Эти программы не только позволяют использовать при составлении программы мнемонику языка ассемблера PDP-11, но и выполнять ряд полезных операций по отладке программ, включая визуализацию содержания регистров общего назначения, пересылку текста программы с одного адреса на другой, пошаговое выполнение программы, сброс отлаженной части программы на ленту и др.

Важным инструментом разработчика программ в машинных командах являются реассемблеры, облегчаю-

щие расшифровку цифрового кода команд. В распоряжении секции имеется 6 программ реассемблирования (АНТИАС, КОДЫ, КОД 24, REASS, ASC11, MESSA).

Небольшая, но весьма мощная система FOCOD, разработанная В. Н. Андреевым, позволяет объединить программы, написанные на Фокале, с программами в машинных командах. По сравнению с Бейсиком «Электроника 60» разработка программ в машинных командах и обращение к ним в Фокоде предельно упрощены. Объединение Фокода с модифицированной программой записи содержимого экранной памяти на магнитную ленту и считывания с ленты в ОЗУ позволило разработать простой графический экраный редактор, существующий в двух модификациях (FOCODMAG3 и GRAFRED).

Для работы над текстовыми материалами с помощью БК-0010 предназначены несколько текстовых экраных редакторов. К ним относятся программа БЛОКНОТ, входящая в систему РИГА, а также программы SOM и EKRED. Последняя программа, разработанная с использованием Фокода, дает возможность одновременно обрабатывать тексты объемом до 8 страниц по 1534 символа на странице. Программа позволяет вводить, стирать или переносить отдельные символы, слова, строки, абзацы, группы абзацев и целые страницы, а также смещать текст на странице вправо и влево целиком. Любой заданный объем можно вывести на магнитную ленту или на принтер. Программа EKRED4 снабжена подпрограммой специального алфавита, позволяющей легко создавать тексты на любом национальном языке.

Полезным дополнением к Фокалу является программа СБОРКА, позволяющая объединять в единую программу отдельные блоки, написанные на Фокале и хранящиеся на магнитной ленте. Стандартный Фокал БК-0010, как известно, не обладает такой возможностью. Реализация сборки фокальных программ из блоков расширяет возможности БК-0010 и сокращает время на разработку прикладного программного обеспечения благодаря использованию тематических библиотек стандартных блоков, из которых, как из кубиков конструктора, можно собрать необходимые программы.

Для составления словарей, каталогов и других текстовых массивов, записи в которых вызываются ключевым словом или группой ключевых слов, разработана программа БАЗА. На ее основе создан англо-русский словарь ANRUS на 700 слов, а также переводчик англо-американских мер в метрические.

В распоряжении пользователей — любителей БК-0010 имеются три программы музыкальных редакторов (БЛОКНОТ, МЕЛОМАН, MUSIC), с помощью которых можно создавать разнообразные музыкальные программы, записывать и воспроизводить на компьютере мелодии, снабжать прикладные и игровые программы звуковым сопровождением, обучать музыкальной грамоте.

Среди других усовершенствований системного обеспечения БК-0010 можно упомянуть программу UPAK упаковки массива чисел, систему интерфейса внешних функций (ИВФ) и программу ИВФ МАССИВЫ, дополняющую систему ИВФ. Программа UPAK позволяет упаковывать число с плавающей запятой в 4 байта вместо восьми и тем самым размещать в ОЗУ до 6500 чисел с произвольным доступом к каждому. Система ИВФ вводит дополнительные внешние функции, подпрограммы для которых могут быть

написаны в машинных командах, а сами функции — использоваться Фокалом. Программа ИВФ МАССИВЫ вводит несколько таких внешних функций, которые удобно использовать при организации за пределами области ОЗУ, оведенной Фокалу, трех отдельных массивов с произвольным доступом — для байтов, целых чисел и чисел с плавающей запятой. Первый из этих массивов нетрудно преобразовать в массив символьных величин, которые отсутствуют в Фокале.

Еще более плотная упаковка информации достигнута в программах научно-технических справочников SPRA, SPPE, SPRI и FOSPR. В этих программах реализован принцип побитного кодирования стандартных текстовых характеристик, который позволяет, в частности, всего в четырех байтах упаковать информацию, занимающую после расфасовки объем более 0,5К байта. Применение подобных программ дает возможность в компьютере со сравнительно небольшой памятью создавать информационные массивы значительного размера, удобные в обращении.

Имеется несколько программ графо-аналитической обработки данных: GIST, GRAPH, UNIGRAF, GRANOD. Наиболее универсальна программа GRANOD, которая позволяет строить график функции по точкам, проводить выравнивание по девяти наиболее распространенным типам аналитических зависимостей с автоматическим вычислением параметров выбранной эмпирической формулы методом наименьших квадратов. С помощью программы UNIGRAF можно на одном графике вычерчивать несколько графиков функций, заданных аналитически.

Значительное число разработанных и адаптированных программ предназначено для выполнения расчетов. В дополнение к программе GIST, поставляемой вместе с компьютером созданы программы POLYNOM, DIF, MNK, DIFUR, COMPLEX, DETER, NEWTON, SIMPSON, STATSB, SOM TRIGINT, DIEHLE, MINIM, BETARAB, GAPMAN, KOREI, MNUGULB, PRESNB, PROCON, PROSTB, SKOLZB. Это — программы вычисления коэффициентов интерполирующего полинома, решения систем условных линейных уравнений методом наименьших квадратов, интегрирования и дифференцирования аналитически и таблично заданных функций статистических расчетов, решения системы дифференциальных уравнений, обращения матриц и др.

Есть программы, обеспечивающие специальные расчеты, в частности, надежности аппаратуры, механических, радиотехнических и электронных приборов и схем, физико-химических и термодинамических констант вещества. Таковы программы ATTENUB, DCCONVERT, BRICKI, GLASS, LOGIKA, MAFLA, NADEGNB, NIGFILB, OBZLENS, POLFILB, REDPLAN, RELIAB, TRANUSB, VERFILB, TERM, MOLMASSA, ENTOBR, EGLOF, TKAMOVOSPL.

Среди программ управления внешними устройствами выделим программы: БЕГСТР для управления установкой лазерной графики, КОНСУЛ для сопряжения БК-0010 с печатающей машинкой, ДМП для управления электронными приборами.

Значительную долю программного обеспечения составляют учебные программы: по арифметике (АРИФМЕТИКА), истории (ДАТЫ), географии (ЛАТВИЯ), иностранному языку (ЗУБЕР), русскому языку (комплекс программ РУС), музыкальной грамоте (НОТЫ), правилам дорожного движения (ПДД), гражданской обороне (ХИМОРУЖИЕ). Ряд обучающих программ предназначен для изучения курса программирования на БК 0010 (ГРАФБЛР, ПУЛЬТ, БК-0010, KLAU), физики (BROWN, ОПТИКА, ЕККАД, ELSLK), информатики и вычислительной техники в школе (ЭВМ, РОБОТ, SHPAGA MURAWEJ), биологии (ЖИЗНЬ, АКВАТОР), геометрии (CUBE, DEKART), для развития композиционных способностей

детей (БК или КОМПЬЮТЕРНЫЙ МИР, ПЕЙЗАЖ, GENERATOR, KRUG). Некоторые обучающие программы, рассчитанные на детей, включают игровые элементы.

Собственно игровые программы представлены наиболее широко и составляют почти половину всех программ.

Среди них программы *логических игр*: шахматы (CHESS, CHESS3 и CHESSM), шашки (REVERSY, SKREPKI, HANKI), домино (ДОМИНО), кубик Рубика (RUBIK) и менее распространенные игры: ЦЯН, KALAH, LOTO, NIM, NIMB, TRIGEXS, CHISLO, GUESSB, TWENTYONE.

Заслуживают внимания программы *деловых игр*. Один тип игр (MANAGER, MANAGEMENT, MNG23) имитирует управление промышленным предприятием. Второй тип (PRZ, KING, DICTAT, VAVILON) — хозяйственно-политическое управление, что пручает игрока к рациональному распределению ресурсов, мышлению экономическими и политическими категориями.

Приключенческие игры (ПУТЕШЕСТВИЕ, OREGON) имитируют насыщенные разными приключениями путешествия по диким берегам Америки, полеты в космос (KOSMOS, LUNARM, KLINGI, KLINGON), приключения в замках, подземельях и лабиринтах (ЗАМОК, КОЛОБОК, ПАКМАН, FARAON, WAMPUS). Обычно эти программы включают элементы динамической графики. Вообще *динамические игры* — это одна из наиболее привлекательных задач программирования для начинающих самостоятельных программистов. Среди них игры с мячом (теннис, бадминтон, футбол, SKIOSH); игры типа посадки на Луну (FMOON, MOON, LUNA, CHELNOK); игры типа удава, поедающего зайцев (PITON, УДАВ, UDAV, KOCA, SNAKEB, OXOTA); защиты от чудовища или преодоления полосы препятствий (ЗАМОК, WUMPUS, КОЛОБОК, ПАКМАН, ASTEROIDB, YAMANA); «стрельбы по цели» (AKTILER, PERES, RANGE, ROBIN, STREL, TIRM, MAPC); гонок или скачек (ГОНКИ, COWBOY, GONKI, SLALOM); сражений с космическими пришельцами, вражескими самолетами, подводными лодками и т. п. (OKEAN, DOGFIGHT, FLIGHTB, KLINGI, KLINGON, SPACEWAR, SPACEW, SPW, MORBOY).

Отдельную группу составляют демонстрационные программы, в которых пользователь выступает в основном в роли зрителя. Обычно это графические программы, иногда сопровождающиеся музыкальными эффектами. Например, в программе ДОМИК маленький робот строит домик, называя каждую его деталь, а закончив строительство, заразительно пляшет под музыку и хвалит построенный домик. Эта веселая программа демонстрирует возможности БК-0010 по графике, синтезу речи и музыки. К программам этой группы относятся также ЖИЗНЬ, ПЕСНИ, АКВАТОР, BANDIT, BIORITMY, CELL, CUBE, LUN, LUNP, GENERATOR, PIXEL, VAPNA, DEMO, REKLAMA, AELITA, WINNI.

К группе психологических тестов относятся программы НЕРВЫ, CHARACTER, ARTIST, SAMOTEST, TREWS, с помощью которых пользователь может определить тип своего темперамента, оценить художественные способности или состояние нервной системы.

По вопросам приобретения программ следует обращаться непосредственно к членам секции пользователей БК-0010, принимающим участие в заседаниях клуба «Интерфейс», которые проводятся, как правило, каждый четвертый четверг месяца с 10 ч в Доме научно-технической пропаганды им. Ф. Э. Дзержинского по адресу: Москва, ул. Кирова, 7.

Статья поступила 10 января 1987 г.

УЧЕБНАЯ ЛОКАЛЬНАЯ СЕТЬ МИКРОЭВМ

В Ждановском металлургическом институте создана локальная сеть микроЭВМ, предназначенная для обучения языку Паскаль и разработки на этом языке эффективных программ для микроЭВМ «Электроника БК-0010» (далее «БК-0010»). Сеть состоит из четырех «БК-0010» и центрального комплекса «Электроника 60М». Возможно подключение до 10-15 «БК-0010». Решены вопросы полноценного переноса программ, построенных с помощью компилятора Паскаля, на «БК-0010», а также доступа пользователей сети к файлам на устройствах центрального комплекса. Работа сети поддерживается операционной системой ОС ДВК и сетевым монитором, дающим возможность пользователям редактировать исходные тексты программ, компилировать их средствами ОС ДВК, загружать

```

    E341 : CMPB R0, #341      ; ОБСЛУЖИВАНИЕ ЗАПРОСОВ НА ВЫВОД СИМВОЛА
           BNE E331         ; («.TTUUD», «TTUUR»).
           MOV #2(CSP), R0
           JSR PC, OUTPUT
           BR EXENT
    20x : CLR FLG
           BR EXENT

    E341 : CMPB R0, #341      ; ОБСЛУЖИВАНИЕ ЗАПРОСОВ НА ВЫВОД СИМВОЛА
           BNE E331         ; («.TTUUD», «TTUUR»).
           MOV #2(CSP), R0
           JSR PC, OUTPUT
           BR EXENT

    .PRINT
    E351 : CMPB R0, #351      ; ОБСЛУЖИВАНИЕ ЗАПРОСА .PRINT.
           BNE EXIT
           MOV #2(CSP), R0
           SEC EXENT
           CMPB (R0), #200
           BEQ EXENT
           MOV #1(R0), R0
           JSR PC, OUTPUT
           BR PRINT

    EXIT : CMPB R0, #350      ; ОБСЛУЖИВАНИЕ ЗАПРОСА .EXIT.
           BNE FAIL
           BNE LT
           FAIL
           LT

    FAIL : MOV #MESSAGE, R1   ; СМЛА ПЕРЕДАЕТСЯ УПРАВЛЕНИЕ, ЕСЛИ
           BR PRINT         ; ОБРАБОТКА ТЕКУЩЕГО ЭНТ НЕ РАБОТАЕТ.

    MESSAGE: .ASCII 'ERROR IN ENT <<12>'
           .EVEN

    EXENT : JSR PC, #0000     ; ЗАБЕРИТЬ ОБРАБОТКУ ПРЕРВАНИЯ
           BIC #1(CSP)
           MOV (CSP), R1
           MOV (CSP), R3
           RUI
           .EOD
    
```

ЗАГРУЗКА
 11:09:06
 АССЕМБЛЕР MACRO-11 / СОСДК
 МОДУЛЬ ST10 ОБЕСПЕЧИВАЕТ РАБОТОСПОСОБНОСТЬ ПРОГРАММЫ
 ПОСТРОЕННАЯ КОМПИЛЯТОРОМ ПАСКАЛЯ, НА БК-0010. МЕТКА XT10 -
 ТОЧКА ВХОДА В ЗАГРУЗОЧНЫЙ МОДУЛЬ, ОБРАЩАЮЩАЯСЯ КРИ. КОМАНДОВКЕ
 ОБЪЕКТНОГО МОДУЛЯ ST9.СД) И МОДУЛЯ ПРОГРАММЫ. ФРАГМЕНТЫ
 ST10 ОБЕСПЕЧИВАЮТ НАЧАЛЬНУЮ НАСТРОЙКУ ИСПОЛНЯЕМОЙ СИСТЕМЫ
 ПАСКАЛЯ И ПРЕОБРАЗУЮТ ЗАПРОСЫ К МОНИТОРУ СОСДК В ЭКВИВАЛЕНТНЫЕ
 ЗАПРОСЫ К МОНИТОРУ БК-0010, РАСПОЛОЖЕННОМУ В ПЗУ.

ОПРЕДЕЛЕНИЯ
 .GLOBAL XRESR5, XRESR6, XCORE, XFREE ; ЯЧЕЙКИ ДЛЯ СВЯЗИ С
 ИСПОЛНЯЕМОЙ СИСТЕМОЙ
 ПАСКАЛЯ.
 .GLOBAL XBEGIN ; ТОЧКА ВХОДА ПРОГРАММЫ
 .BUFFER = 100 ; ПОСТРОЕННАЯ КОМПИАТОРОМ
 РАЗМЕР БУФЕРА ДЛЯ ПРИЕМА
 NETLOW = 34000 ; СТРОКИ С КЛАВИАТУРЫ
 ; ВЕРХНИЙ АДРЕС ДОСТИЖИМОЙ
 ; ПРОГРАММЕ ПАМЯТИ.

XT10 ::
 ENT 4 ; ИНИЦИАЛИЗИРОВАТЬ КРАЙНЕР КЛАВИАТУРЫ
 MOV #NETLOW, SP ; БК-0010. УСТАНОВИТЬ УКАЗАТЕЛЬ СТЕКА
 MOV SP, XRESR6 ; НАСТРОИТЬ ЯЧЕЙКИ ИСПОЛНЯЕМОЙ СИСТЕМЫ.
 CLR XFREE
 MOV XRESR5, R5 ; РЕГИСТР R5 УКАЗЫВАЕТ НА НАЧАЛО ОБЛАСТИ
 MOV R5, XCORE ; ГЛОБАЛЬНЫХ ДАННЫХ
 MOV #0, SU00 ; СОХРАНИТЬ ВЕКТОР ПРЕРВАНИЯ ПО КОМАНДЕ
 MOV #0, SU32 ; ENT, ПЕРЕКЛЮЧИТЬСЯ НА ОБСЛУЖИВАНИЕ
 JSR PC, XJT00 ; ЗАПРОСОВ К МОНИТОРУ СОСДК.
 JMP XBEGIN ; ПРИСТУПИТЬ К ИСПОЛНЕНИЮ ПРОГРАММЫ.
 SU30 : .WORD ; ЯЧЕЙКИ ДЛЯ ХРАНЕНИЯ ВЕКТОРА ПРЕРВАНИЯ
 SU32 : .WORD ; ПО ENT.
 BUFFER : .BLKB BUFSTZ ; БУФЕР ДЛЯ ПРИЕМА СТРОКИ С КЛАВИАТУРЫ.
 ENDBUF : .WORD
 DELIM : .BYTE BUFSTZ
 .BYTE 12
 COUNT : .WORD ; ПРЕДЕЛЬНОЕ ЧИСЛО СИМВОЛОВ В СТРОКЕ.
 FLG : .WORD 0 ; ОГРАНИЧИТЕЛЬ СТРОКИ.
 RUS : .WORD 0 ; УКАЗАТЕЛЬ НА СИМВОЛ, ПРОЧИТАННЫЙ ИЗ
 ; БУФЕРА. ЯЧЕЙКА FLG ОЧИЩАЕТСЯ, КОГДА
 ; ПРОЧИТАНЫ ВСЕ СИМВОЛЫ.
 ; ПРИЗНАК ТЕКУЩЕГО РЕГИСТРА («RUS ИЛИ LAT»)

XJT00 : MOV SU30, #030 ; ПОДПРОГРАММА ПЕРЕКЛЮЧЕНИЯ НА ОБСЛУЖИВАНИЕ
 MOV SU32, #032 ; ЗАПРОСОВ К МОНИТОРУ БК-0010.
 RTS PC
 XJT03 : MOV #ENT00, #030 ; ПОДПРОГРАММА ПЕРЕКЛЮЧЕНИЯ НА ОБСЛУЖИВАНИЕ
 CLR #032 ; ЗАПРОСОВ К МОНИТОРУ СОСДК.
 RTS PC
 OUTPUT : CMPB R0, #16 ; ПОДПРОГРАММА ВЫВОДА СИМВОЛА С УЧЕТОМ
 BNE 1x ; ПРИЗНАКА РЕГИСТРА («ЯЧЕЙКА RUS»).
 MOV #1, RUS
 RTS PC
 1x : CMPB R0, #17 ;
 BNE 2x
 CLR RUS
 RTS PC
 2x : TST RUS
 BEQ 3x
 CMPB R0, #77
 BLS 3x
 BVS #000, R0
 ENT 10 ;
 RTS PC
 3x :
 ENT 10 ;
 RTS PC
 ENT00 : MOV R0, -(CSP) ; ОБРАБОТКА ПРЕРВАНИЯ - СОХРАНИТЬ
 MOV R1, -(CSP) ; РЕГИСТР, ИЗВЛЕЧЬ КОД КОМАНДЫ ENT,
 MOV #CSP, R0 ; ВЫВОДИТЬ ПРЕРВАНИЕ, ПРИСТУПИТЬ К
 MOV -(R0), -(CSP) ; АНАЛИЗУ КОДА.
 MOVS (CSP), R0
 JSR PC, #R10
 CMPB R0, #340 ; ОБСЛУЖИВАНИЕ ЗАПРОСОВ НА ВВОД С
 BNE E341 ; КЛАВИАТУРЫ («.TTUUD», «TTUUR»).
 TST FLG
 BNE 10x
 MOV R2, -(CSP)
 MOV #BUFFER, R1
 MOV DELIM, R2

программы в «БК-0010» для выполнения или записи на магнитную ленту битовым магнитофоном.

С целью уменьшения аппаратных затрат на создание сети разработана схема соединений. «Электроника 60М» подключена к сети через параллельный интерфейс И2, а «БК-0010» — через порты ввода-вывода. Скорость передачи информации 250 16-разрядных слов/с.

Сетевой монитор состоит из программы MISC, исполняемой в центральном комплексе под управлением ОС ДВК, и экземпляров программы NET, исполняемых в «БК-0010». NET проверяет синтаксис принимаемых с клавиатуры команд сетевого монитора и включает в себя экранный редактор текстовых файлов. MISC выполняет файловые операции, последовательно опрашивая готовые к взаимодействию экземпляры NET.

Команда сетевого монитора задается в виде строки символов, завершаемой символом ВВОД. Общий формат команд

[КОМ] [ИМЯ ФАЙЛА [ТИП ФАЙЛА]]

где КОМ — односимвольное имя команды; квадратные скобки указывают на необязательное присутствие элемента, заключенного в них.

Допустимо опускать тип файла, при этом он назначается автоматически исходя из имени команды и номера пользователя, введенного в начале работы. Например, пользователь 15 модифицирует программу с именем FILE. По команде FILE в память «БК-0010» загружается файл FILE.P15 с исходным текстом программы и устанавливается режим экранного редактирования. Новая версия файла записывается после возврата к режиму приема команд командой FILE. Команда @COMP запускает командный файл COMP.C15, в котором запрограммирована компиляция FILE.P15. Созданный при компиляции листинг LIST.P15 можно посмотреть, подав команду LIST. При выполнении командного файла MAC.C15 (команда @MAC) получаем файл FILE.S15, содержащий готовую программу в формате отображения памяти. По команде >FILE программа загружается в память БК-0010 и запускается.

Каждый пользователь имеет набор необходимых ему командных файлов. Возможностей, предоставляемых

файлами COMP.C15 и MAC.C15, достаточно для начального обучения Паскалю.

Файл COMP.C15: R PASCAL
FILE.M15,LIST.P15=FILE.P15/F
Файл MAC.C15: R MACRO
FILE.J15 = FILE.M15
7C
R LINK
FILE.S15=SY:SY10,DK:FILE.J15/T
*ST10
7C

Чтобы программа, написанная на Паскале и предназначенная для работы под управлением ОС ДВК, могла выполняться на БК-0010, был создан модуль ST10 — ловушка прерываний по команде EMT. Этот модуль присоединяется к модулям программы на этапе компоновки и преобразует запросы на ввод-вывод, адресованные к монитору ОС ДВК, в эквивалентные запросы к монитору «БК-0010», расположенному в ПЗУ.

УДК 681.325—181.4

А. М. Петух, А. Н. Романюк, Д. Т. Ободник, О. А. Подольский

ПРОГРАММНО-АППАРАТНАЯ РЕАЛИЗАЦИЯ ЛИНЕЙНОГО ИНТЕРПОЛЯТОРА

Повысить эффективность микропроцессорных интерполяторов можно, оптимально распределив аппаратные и программные функции и введя дополнительные функциональные узлы, реализующие функции, программное выполнение которых требует больших временных затрат.

При программно-аппаратной реализации двухуровневого метода линейной интерполяции на верхнем уровне программным путем определяются многоадресные приращения, соответствующие цифровым сегментам аппроксимируемого отрезка прямой. На нижнем уровне графический интерпретатор аппаратным путем преобразует полученные многоадресные приращения в управляющие сигналы индикаторных или регистрирующих устройств.

Цифровые сегменты, из которых состоит заданный отрезок прямой, определяет микропроцессор (микроЭВМ «Электроника 60М»), вычисляя неравномерные характеристики первого порядка. При обработке графическим интерпретатором многоадресных приращений, поступающих из микропроцессора, генерируется отрезок прямой.

Поскольку вектор аппроксимируется цифровыми сегментами двух длин, отличающихся на единицу, то алгоритм функционирования микропроцессора в режиме линейной интерполяции достаточно прост и сводится к комплексному учету неравномерных характеристик более высокого порядка.

В цикле подготовки к интерполяции микропроцессор анализирует при-

ращения ΔX и ΔY , задающие отрезок прямой для определения его области расположения и направления, а также вычисляет цифровые сегменты, определяющие структуру заданного вектора.

Последовательности, включающие одиночный, $K+1$ и K импульсов, поступают на входы двоечного мультиплексора D9, коммутирующего (в зависимости от состояния управляющего слова, записанного в регистр D6) соответствующие заданному отрезку прямой частотные последовательности для координат J и Y .

Частотные потоки, поступающие с выхода коммутатора D9, распределяются элементами D10 на выходы $+X$, $-X$, $+Y$, $-Y$ интерполятора в зависимости от знака приращений, задающих исходный отрезок.

Разрядность счетчика D7 зависит от размеров дискретного координатного пространства.

Выходной сигнал L служит сигналом готовности графического интерпретатора к приему нового многоадресного приращения от микропроцессора, или сигналом запроса прерывания процессора.

В цикле интерполяции микропроцессор выдает на графический интерпретатор требуемое значение одного из двух возможных многоадресных приращений. При отсутствии повышенных требований к точности аппроксимации очередной цифровой сегмент может определяться суммированием в каждом интерполяционном такте остатка от деления, полученного при определении K , по модулю делителя. При переполнении

Так, запрос на вывод символа EM1341, код которого расположен в регистре R0, принимается ST10 и преобразуется в запрос EMT16 аналогичного назначения.

Сеть применяется в учебном процессе в курсах программирования и АСУ ТП. Сетевой монитор может быть использован в условиях класса промышленного исполнения.

Адрес для справок: 341004, Донецкая область, г. Жданов, ул. Заозерная, 98-б, Водяник А. Г.

ЛИТЕРАТУРА

1. Математическое обеспечение микроЭВМ «Электроника БК-0010». Руководство системного программиста. Т. 1, кн. 2.—1985.
2. Ги К. Введение в локальные вычислительные сети. — М.: Радио и связь, 1986.
3. Комплектный класс технических средств на базе микроЭВМ «Электроника БК-0010Ш» и ДКВ-2МШ / Г. И. Фролов, С. М. Косенков, В. А. Шахнов и др. // Микропроцессорные средства и системы.—1986.— № 4.

Статья поступила 15 ноября 1986 г.

сумматора выдаче подлежит сегмент длиной $K+1$ (в противном случае — длиной K).

Алгоритм обработки отрезков прямых (с минимальным отклонением от идеальной прямой, равным половине шага дискретизации) и точного попадания в конечную точку сводится к шести операциям.

Быстродействие предлагаемого линейного интерполятора более высокое по сравнению с программной реализацией методов ЦДА и оценочной функции, поскольку прямые аппроксимируются многоадресными приращениями, существенно сокращено время анализа попадания в конечную точку; значительно расширены возможности организации обмена данными в режиме прерывания программы; обеспечена параллельная работа процессора и внешних исполнительных устройств.

Статья поступила 20 ноября 1985 г.

Адрес для запроса схем и программ; 286021, Винница, Хмельницкое шоссе, 133. Винницкий политехнический институт,

АЦП НА БИС К1113ПВ1 ДЛЯ ПЕРСОНАЛЬНОЙ ЭВМ «ЭЛЕКТРОНИКА БК-0010»

Широкое использование ПЭВМ «Электроника БК-0010» сдерживается среди прочих причин отсутствием в комплекте поставки устройств связи с объектом, в качестве которых обычно используют АЦП, организованные в многоканальные системы. Серийные АЦП, например «Электроника 15КА-60/8-010» и «Электроника 15ПРА-13-001», сложны, недостаточно надежны и по стоимости превышают ПЭВМ. В настоящее время АЦП выпускаются в виде БИС и представляют собой функционально законченные устройства. Это избавляет потребителя от необходимости разрабатывать АЦП, но требует решения вопросов по объединению их в систему.

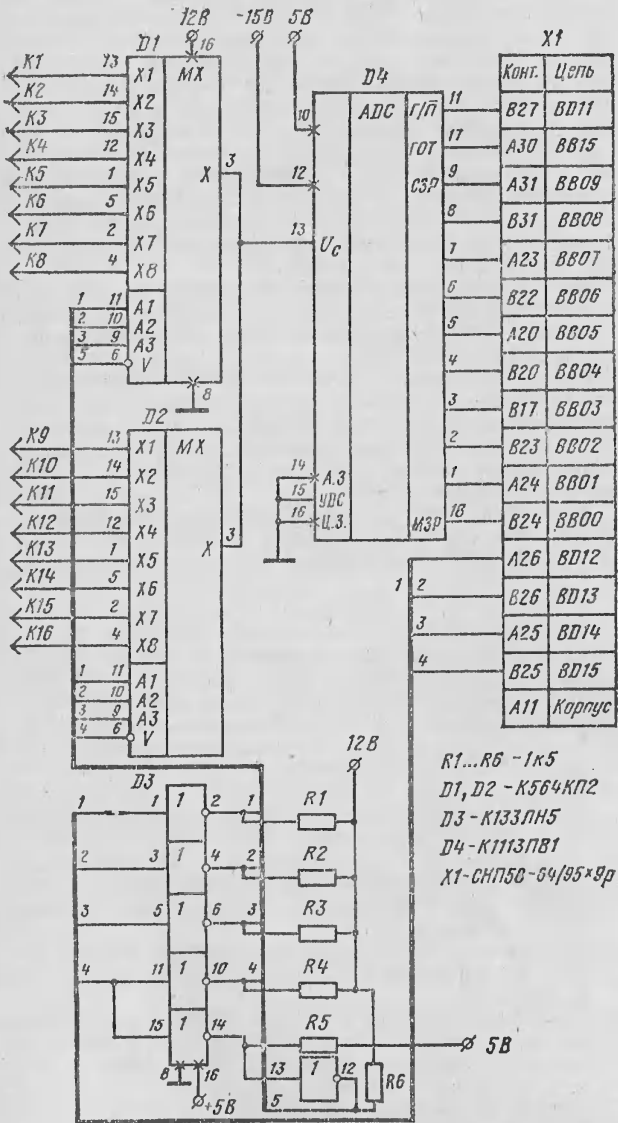


Рис. 1. Принципиальная схема 16-канального 10-разрядного АЦП

Примером функционально законченного АЦП может служить микросхема К1113ПВ1, предназначенная для использования в составе устройств аналогового ввода [1]. Микросхема выполняет функцию 10-разрядного АЦП однополярного или биполярного входного сигнала с представлением результата в параллельном двоичном коде и содержит источник опорного напряжения, компаратор и тактовый генератор. Наличие выхода с тремя состояниями позволяет подключать к одной шине несколько БИС для получения многоканальной системы. Для экономии числа БИС К1113ПВ1 многоканальный режим можно организовать с помощью коммутаторов, подключенных ко входам АЦП. Разработанный на основе БИС К1113ПВ1 преобразователь включает минимальный набор микросхем. В качестве входного коммутатора используется КМОП БИС К564КП2, что позволяет поднять верхнюю границу $U_{\text{вх. макс}}$ до 10 В и обеспечить время преобразования, равное 30 мкс [2].

Ввиду того, что большинство интерфейсных шин современных ЭВМ, в частности ПЭВМ «Электроника БК-0010», имеют уровни «Лог.0» и «Лог.1», соответствующие уровням ТТЛ ИС, требуется установка элементов согласования ТТЛ — КМОП К133ЛН5.

Сигналы от измерительных преобразователей поступают на входы X1...X8 коммутаторов D1, D2 (рис. 1). Номер канала задается по шине ПЭВМ в виде 4-разрядного слова и поступает на элемент согласования D3, на выходе которого появляются управляющие сигналы, соответствующие номеру подключаемого к АЦП канала.

Используя совместимость системы команд ПЭВМ «Электроника БК-0010» с микроЭВМ «Электроника 60», программу обращения к АЦП можно заимствовать из программного обеспечения указанных ЭВМ после соответствующей доработки. В тех случаях, когда не требуется высокого быстродействия, можно использовать возможности резидентного программного обеспечения микроЭВМ «Электроника БК-0010».

Пример программы обращения к четырем каналам АЦП на языке Фокал приведен на рис. 2. С учетом высокой инерционности интерпретатора в данной программе не анализируется готовность АЦП. Необходимая задержка формируется введением дополнительного пробела в текст программы между командой запуска и считывания данных.

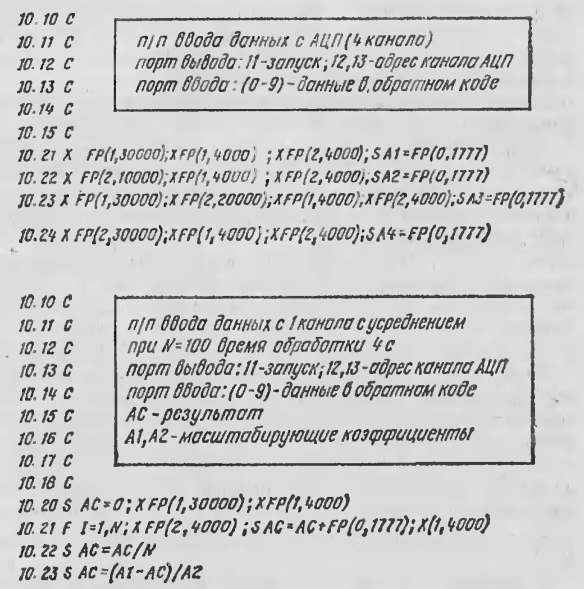


Рис. 2. Программа обращения к четырем каналам АЦП

А. П. Казанцев

ИНТЕРФЕЙС ВНЕШНИХ ФУНКЦИЙ
ИНТЕРПРЕТАТОРА ФОКАЛ-БК-0010

При непосредственном соединении выходов нескольких микросхем К1133ВН1 полностью исключается взаимное влияние аналоговых каналов друг на друга, а их выборка обеспечивается путем дешифрации входов «запуск» АЦП. Такое подключение удобно при работе в условиях сильных промышленных помех. Питание схемы осуществляется от источников напряжений 5, 12 и -15 В с допустимыми отклонениями от номинала $\pm 10\%$. Для снижения уровня помех по цепям питания рекомендуется в местах их подвода к АЦП установить конденсаторы емкости не менее 100 мкФ на соответствующие напряжения.

Микросхема типа К133ЛН5 может быть заменена на К155ЛН5, а К564КП2 после незначительных схемных изменений на К564КП1 или К561КП1. При использовании ограниченного числа каналов АЦП следует недействующими входы D1 и D2 заземлить.

Указанные схемные решения были реализованы в аппаратно-программном комплексе системы управления технологическим процессом производства технического стекла.

Адрес для справок: 198013, Ленинград, Московский проспект, 26. ЛТИ им. Ленсвета, каф. автоматизации.

ЛИТЕРАТУРА

1. Бытовая персональная ЭВМ «Электроника БК-0010» // С. М. Косенков, А. Н. Полосин, З. А. Сцепицкий и др. // Микропроцессорные средства и системы.—1985.— № 1.— С. 22—25.
2. Федорков Б. Г., Телец В. А., Дегтярков В. П. Микроэлектронные цифро-аналоговые и аналого-цифровые преобразователи.— М.: Радио и связь, 1984.

Статья поступила 26 января 1987 г.

(Окончание. Начало см. на стр. 61)

рует работу микрокалькулятора, снабженного аккумулятором (переменная АККУ) и стековой памятью, реализованной с помощью семи регистров (переменные А, В, С, D, F, G). Промежуточные и окончательный результаты вычислений запоминаются в аккумуляторе. Первый операнд в арифметических действиях выбирается из АККУ, а второй — из верхушки стека (переменная А). После выполнения операции содержимое стека «поднимается». При вычислении значений стандартных функций аргументом является содержимое АККУ, сюда же заносится и результат. Если стек исчерпан, то можно выполнить только одностороннюю операцию над содержимым аккумулятора. Ответ, полученный в результате вычислений по заданной формуле, может быть использован в последующих расчетах.

Беглый анализ программы «школьная математика» показывает, что она содержит очнь много стереотипных фрагментов и за счет этого имеет довольно большой объем — 208 строк (около 250 операторов). Значительное сокращение этой программы (почти в четыре раза) может быть достигнуто за счет двух следующих соображений.

Во-первых, вместо простых переменных, отведенных под стек и аккумулятор, можно использовать одномерный массив А, содержащий 8 элементов ($A(0) = АККУ$, $A(1) = А$, $A(2) = В$, ..., $A(7) = G$). Это позволит вводить числовые данные в цикле по единой процедуре.

Во-вторых, существенное сокращение объема программы может быть достигнуто за счет объединения стереотипных фрагментов, в которых выводится наименование выполняемой операции и анализируется ответ пользователя, управляющего запуском операции. С этой целью в состав программы необходимо ввести символичный массив ВJ, элементами которого ВJ(J) являются наименования операций, соответствующие числовому коду J.

В конечном итоге, предлагаемые приемы позволяют сократить объем исходной программы до 50 строк.

Система FOCAL-11 имела развитый программный интерфейс внешних функций, обеспечивавший профессиональные применения наиболее простых и дешевых конфигураций систем PDP-11 и отечественных систем, совместимых с ними. В настоящее время продано более 4 тыс. микроЭВМ «Электроника БК-0010», оснащенных подобным интерпретатором (Фокал-БК-0010), однако, лишенным указанного интерфейса.

В Институте биологической физики АН СССР разработан интерфейс внешних функций (ИВФ) интерпретатора Фокал микроЭВМ «Электроника БК-0010» для применения этой популярной машины в автоматизации научных исследований. ИВФ (0,5К байт ОЗУ) позволяет встраивать коды новых функций в Фокал-БК-0010. Эти функции готовятся на ассемблере или в кодах самой «Электроники БК-0010» (ДВК, «Электроника 60», СМ-4) с использованием развитых инструментальных средств. Внешние функции могут входить в выражения на языке Фокал, разрешено их рекурсивное использование. Для передачи параметров и результатов, для внутренних вычислений и дополнительной интерпретации можно использовать внутренние подпрограммы и функции Фокала. Способ работы с внутренними подпрограммами и функциями описывается в различных системных руководствах по Фокалу других машин (например, в руководстве по диалоговой системе ДС СМ ИНЭУМ для СМ-3, СМ-4).

ИВФ обеспечивает также подключение драйверов устройств вывода символической информации (вместо драйвера экрана), вызываемых при работе операторов Фокала TYPE и WRITE.

Интерфейс внешних функций — удобное средство решения системотехнических задач, доступное даже для начинающих специалистов. Его можно использовать на «Электронике БК-0010», применяемой для автоматизации аналитических приборов, технологических установок и учебных стендов.

Адрес для справок — 142292, Московская обл., г. Пущино, Институт биологической физики АН СССР, Тел. — 923-96-68 (доб. 2-93).

ЛИТЕРАТУРА

1. Вернер В. Д., Воробьев Н. В., Горячев А. В. и др. Микропроцессоры. Средства сопряжения. Контролирующие и информационно-управляющие системы.— М.: Высшая школа, 1986.—383 с.
2. Писаревский А. Н., Осетинский Л. Г., Осетинский М. Г. Фокал-диалоговый язык для мини-ЭВМ.— Л.: Машиностроение, 1986.—195 с.
3. Фролов Г. И., Гембицкий Р. А. Микропроцессоры. Автоматизированные системы контроля объектов.— М.: Высшая школа, 1984.— 87 с.
4. Казанцев А. П., Пронин Л. А. Типовые микро-ИВК и автоматизированные рабочие места исследователей для биологических лабораторий // УСИМ.— 1983.— № 2.— С. 102—106.
5. Казанцев А. П. Автоматизированное рабочее место исследователя // Автометрия.—1982.— № 1.— С. 104—107.
6. Программное обеспечение СМ ЭВМ. Операционная система с разделением функций РАФОС. Диалоговая система программирования ДИАСП. Описание языка, т. 5, кн. 5, ч. 3, 1980.—79 л.
7. FOCAL-11 User's Manual, Digital Equipment Corp., Maynard, Massachusetts, USA, 1975, p. 228.

Статья поступила 11 января 1987 г.

СЕРИЙНЫЕ МИКРОПРОЦЕССОРНЫЕ КОМПЛЕКТЫ БИС

Первые зарубежные и отечественные микропроцессоры были изготовлены по р-МОП-технологии, отличающейся принципиальным ограничением по быстродействию. Поэтому, несмотря на простоту и низкую стоимость таких микросхем, р-МОП-технология, по-видимому, будет использоваться в основном для создания изделий бытовой техники (К145, КР1814).

п-МОП-технология позволила на порядок по сравнению с р-МОП увеличить быстродействие. Эта технология обеспечивает высокую плотность упаковки и быстродействие, сравнимое с ТТЛШ-приборами. п-МОП-технология стала основной для МПК среднего быстродействия, где определяющим требованием является минимальное число кристаллов (КР580, К1801, КМ1810).

КМОП — перспективная технология, не имеющая конкуренции при разработке маломощных систем. Кроме того, МПК, выполненные по этой технологии, имеют более высокую помехоустойчивость и надежно работают в более широких диапазонах напряжений питания и температур окружающей среды (КР588).

ТТЛ-схемы наиболее популярны у потребителей благодаря высокому быстродействию, низкой стоимости, широкой номенклатуре, стабильности параметров. К преодолению недостаткам следует отнести малую плотность упаковки, большую рассеиваемую мощность. Рассеиваемая мощность уменьшается при использовании диодов Шотки — ТТЛШ-технология. МПК БИС в основном создаются по маломощной ТТЛШ-технологии, позволяющей получать высокоскоростные контроллеры периферийных устройств с широкими технико-экономическими характеристиками, а также эмулировать архитектуру существующих ЭВМ (К589, КР1802, КМ1804).

ЭСЛ-технология наиболее трудоемкая и дорогая. Микросхемы, выполненные по этой технологии, имеют максимальное быстродействие, но малую плотность упаковки и большую рассеиваемую мощность. Они могут применяться только с принудительным обдувом, однако незаменимы там, где требуется высокое быстродействие. Эта технология позволяет перевести основную структуру больших универсальных ЭВМ на микропроцессорные ИС.

Необходимо отметить, что все ви-

ды технологий, применяемых в МПК, непрерывно совершенствуются: снижается потребляемая мощность и увеличивается плотность упаковки bipolarных МПК, возрастает быстродействие МПК, выполненных по МОП-технологии. В зависимости от требований, предъявляемых к МПК, в него могут входить микросхемы, выполненные по различным технологиям (п-МОП и ТТЛШ, ЭСЛ и ТТЛШ и т. д.).

Основные характеристики МПК БИС. Микропроцессор — это цифровое устройство обработки информации, выполненное в виде одной или нескольких интегральных микросхем, поэтому ему свойственны параметры как вычислительных устройств (система команд, объем адресуемой памяти и т. д.), так и микросхем (уровни сигналов, мощность рассеяния, помехоустойчивость и т. д.).

В связи с большим разнообразием МП и МПК (универсальные и специализированные, однокристалльные, многокристалльные и секционные, синхронные и асинхронные, одномагистральные и многомагистральные и т. д.) определить единую систему параметров, позволяющую оценить технические возможности МПК, довольно сложно, поэтому рассмотрим основные из них, дающие ориентировочную оценку.

Разрядность обрабатываемых данных — параметр, определяющий точность вычислений. Существуют МП с фиксированной (4, 8, 16) и наращиваемой разрядностью.

В МП с фиксированной разрядностью (КР580, КР588, К1801, КМ1810) увеличение разрядности обрабатываемых данных возможно путем просчета в несколько этапов, что снижает быстродействие систем.

В МП с наращиваемой разрядностью (К589, К1800, КР1802, КМ1804) микропроцессор строится из нескольких микропроцессорных секций, каждая из которых имеет n разрядов. Тогда разрядность обрабатываемых данных определяется как $n \times K$, где $n = 2, 4, 8, \dots$; $K = 1, 2, 3, \dots$ (число секций).

Система команд — совокупность операций, обеспечивающих выполнение программы в соответствии с заданным алгоритмом. В это понятие входят: формат команд и обрабатываемых данных, число команд, способы адресации данных, объем непосредственно адресуемой памяти, возможности стека, способы обработки

прерываний, организация ввода-вывода и т. д., поэтому сравнение МП по числу выполняемых команд может привести к ошибочным результатам. Необходима качественная оценка системы команд выбранного микропроцессора. Кроме того, система команд определяет программное обеспечение системы, разработка которого для мощных вычислительных и управляющих систем составляет значительную часть стоимости разработки всей системы. Когда разрабатываемое микропроцессорное устройство используется на нижних уровнях сбора и обработки информации с последующим согласованием с ЭВМ более высокого уровня, тогда при выборе МПК прежде всего необходимо учитывать совместимость по системе команд МП с фиксированной разрядностью имеет фиксированную систему команд. Причем МП серий КР588 и К1801 ориентированы на систему команд микроЭВМ «Электроника 60», а МП серий КР580 и КМ1810 на систему команд микроЭВМ СМ1800 и СМ1810.

МП с наращиваемой разрядностью (секционированные) ориентированы на микропрограммное управление и позволяют пользователю в зависимости от специфики разрабатываемого устройства создавать собственные системы команд. Применение МПК с микропрограммным управлением наиболее целесообразно при разработке систем специализированного назначения, когда путем создания насыщенных и компактных команд можно достичь высокого быстродействия и существенной экономии памяти программ.

Быстродействие МПК определяется схемно-технологическими возможностями и архитектурными особенностями. При оценке быстродействия МП (МПК) необходимо учитывать, что простое сравнение длительностей машинного такта (командного цикла) может привести к неверным выводам, так как некоторые команды выполняются более чем за один такт. Кроме того, одна и та же команда на разных микропроцессорах выполняется за различное число тактов. Быстродействие универсальных МП в основном определяется числом выполняемых операций (регистр — регистр; регистр — память, сложение, умножение и т. д.) в секунду.

Наиболее комплексной оценкой, позволяющей выбрать оптимальный МПК для конкретного применения, является способ эталонного программирования. Он заключается в том, что для заранее определенного набора эталонных задач проектируемой микропроцессорной системы производится пробное программирование для МПК различных типов,

МПК	Технология	Функциональный аналог	Технические характеристики			Число БИС на 1986 г.
			Разрядность	Быстродействие, тыс. операций/с	Система команд	
КР580	п — МОП ТТЛШ	МСС80	8	500	СМ1800	8 7
КР588	КМОП	—	16	300	«Электроника 60»	6
К589	ТТЛШ	13000	2	1000	Произвольная	8
К1800*	ЭСЛ	М10800	4	3000	Произвольная	8
К1801	п — МОП	—	16	500	«Электроника 60»	6
КР1802	ТТЛШ	—	8	800	Произвольная	10
КМ1804	ТТЛШ	Ам2900	4	800	Произвольная	18
КМ1810	п — МОП	МСС86	16	2500	СМ1810	5

*) Примечание. Комплект К1800 рассчитан на работу в диапазоне температур —10...+75°С. Для остальных МПК верхний предел +70°С.

Нужный МПК определяется на основе анализа времени выполнения эталонного пакета задач, затрат на программирование, требуемого объема памяти.

Потребляемая мощность — параметр, зависящий от схемно-технологического исполнения МПК. При его оценке необходимо учитывать, что в зависимости от технологии изготовления МП может быть выполнен в виде одной или нескольких микросхем, поэтому потребляемая мощность определяется при условии выполнения одинаковых функций.

Разработка и выпуск МПК в стране продолжаются более 10 лет. Некоторые комплекты разработки 70-х годов, вероятно, целесообразно использовать в новой аппаратуре. Наиболее перспективные МПК, предназначенные для применения в аппаратуре народного хозяйственного назначения, представлены в таблице.

Состав и основные технические характеристики МПК

МПК БИС серии КР580 предназначен для построения систем цифровой обработки информации среднего быстродействия, микроЭВМ и контроллеров самого разнообразного назначения.

Отличительной чертой комплекта являются фиксированная разрядность (8 разрядов) и система команд (совместима с микроЭВМ СМ 1800), что однозначно определяет структуру устройств, построенных на его основе.

Часть микросхем комплекта

(КР580ГФ24, КР580ВК28, КР580ВК38, КР580ИР82, КР580ИР83, КР580ВА86, КР580ВА87) выполнена на bipolarной технологии (ТТЛШ), остальные — по п-МОП-технологии.

Микросхемы серии КР580 выпускаются в прямоугольных пластмассовых корпусах по ГОСТ 17467-79:

2123.40-2	— КР580ВМ80А, — КР580ВВ55А, КР580ВТ57, КР580ВГ75, КР580ВВ79;
2121.28-4	— КР580ВК28, КР580ВК38;
2121.28-5	— КР580ВВ51А, КР580ВН59;
2120.24-3	— КР580ВВ53;
21400.20-1	— КР580ИР82, КР580ИР83, КР580ВА86, КР580ВА87;
283.16-2	— КР580ГФ24.

Питание микросхем:

12 В ± 5 %;	5 В ± 5 %;	— 5 В ± 5 %
12 В ± 5 %;	5 В ± 5 %	— КР580ВМ80А
12 В ± 5 %;	5 В ± 5 %	— КР580ГФ24
5 В ± 5 %	— для всех остальных микросхем	

Центральное процессорное устройство параллельной обработки данных КР580ВВ80А

Разрядность обрабатываемых данных	8
Число команд	78
Максимальный объем адресуемой памяти, К байт	64
Число адресуемых устройств ввода-вывода	256/256

Число уровней прерывания	8
Биды адресации	прямая, косвенная, непосредственная, регистровая, по указателю стека
Быстродействие, тыс. операций/с	625 (тип регистра)
Тактовая частота, МГц, не более	2,5
Потребляемая мощность, мВт, не более	1250

Программируемый синхронно-асинхронный передатчик КР580ВВ51А

Длина передаваемых и принимаемых символов, бит	5...8
Число подключаемых внешних устройств	1
Число программируемых режимов работы	5
Максимальная скорость обмена; К бит/с	
синхронный режим	64
асинхронный режим	9,6
Тактовая частота, МГц, не более	2
Потребляемая мощность, мВт, не более	500

Программируемый таймер интервалов КР580ВВ53

Число каналов	3
Число разрядов канала	16
Разрядность шины данных	8
Число программируемых режимов работы	6
Максимальный счет при работе счетчиков в режиме:	
двоичного счета	2 ¹⁶
двоично-десятичного счета	10 ⁴
Тактовая частота, МГц, не более	2,6
Потребляемая мощность, мВт, не более	700

Программируемый параллельный интерфейс КР580ВВ55А

Число каналов (при разрядности канала 8 бит)	3
Число программируемых режимов работы	3
Тактовая частота, МГц, не более	2,5
Потребляемая мощность, мВт, не более	600

Программируемый контроллер прямого доступа к памяти КР580ВТ57

Число каналов прямого доступа к памяти	4
Число разрядов адреса	16
Разрядность шины данных	8
Максимальная длина массива обмена данными, К байт	16
Число программируемых режимов работы	6
Тактовая частота, МГц, не более	3,1
Потребляемая мощность, мВт, не более	600

Программируемый контроллер прерываний КР580ВН59

Число обслуживаемых запросов прерывания	8
Число программируемых режимов работы	3
Время выдачи сигнала прерывания на сигнал «запрос», нс, не более	500
Разрядность шины данных	8
Потребляемая мощность, мВт, не более	500

Генератор тактовых сигналов КР580ГФ24

Число выходных высокоуровневых тактовых сигналов	2
--	---

Максимальная частота выходных тактовых сигналов, МГц, не более 3
 Потребляемая мощность, мВт, не более 755

**Буферный регистр
 КР550ИР82/КР560ИР83 ***

Число разрядов регистра 8
 Время передачи информации от входа до выхода, нс, не более 30 (ИР82)
 22 (ИР83)
 Потребляемая мощность, мВт, не более 800

**Шинные формирователи
 КР580ВА86/КР580ВА87 ****

Число формирователей 8
 Время передачи информации от входа до выхода, нс, не более 30 (ВА86)
 22 (ВА87)
 Потребляемая мощность, мВт, не более 800 (ВА86)
 750 (ВА87)

**Контроллер электронно-лучевой трубки
 (ЭЛТ) для видеотерминалов микроЭВМ КР580ВГ75**

Разрядность шины данных 8
 Число основных команд 8
 Емкость ЗУ на 1 знакоряду 2x80 знаков по 8 бит
 2x16 знаков по 7 бит
 Емкость стеков построчная
 Вид развертки
 Программируется:
 число знаков в знакоряду на экране ЭЛТ 1...80
 число знакорядов в кадре 1...64
 число строк телевизионного раstra в знакоряду тактовая частота, МГц, не более 1...16
 3,25
 потребляемая мощность, мВт, не более 800

**Контроллер клавиатуры и индикации
 КР580ВВ79**

Разрядность шины данных 8
 Емкость ОЗУ отображения, бит 16x8
 Число программируемых режимов по вводу 3
 по выводу 1
 Тактовая частота, МГц, не более 2
 Потребляемая мощность, мВт, не более 600

**Системный контроллер
 КР580ВК28/КР560ВК38**

Разрядность шины данных 8
 Время задержки распространения от входа до выхода, нс, не более 60
 Потребляемая мощность, мВт, не более 1000

МПК БИС серии КР588 выполнен на основе низкопороговой КМОП-технологии и является комплектом среднего быстродействия и минимальной потребляемой мощности. На основе этого МПК целесообразно строить микропроцессорные системы, имеющие автономное питание.

Процессор, включающий арифметическое устройство (КР588ВС2), устройство управления, выполненное на пяти микросхемах КР588ВУ2 (кодировки 0001...0005), и системный контроллер КР588ВГ1, имеет фиксированную систему команд, совместимую

* Для ИС КР580ИР83 выходы с инверсией.
 ** Для ИС КР580ВА87 выходы с инверсией.

с системой команд микроЭВМ «Электроника 60», и обеспечивает: обработку 16-разрядных данных; ввод-вывод байтов и 16-разрядных слов;

обработку прерываний; прямой доступ к памяти; управление магистральными приемопередатчиками.

Микросхемы серии КР588 выполнены в прямоугольных пластмассовых корпусах:

- 22.06.42-2 — КР588ВС2, КР588ВУ2, КР588ВГ1;
- 2121.28-4 — КР588ИР1, КР588ВА1;
- 2107.18-2 — КР588ВГ2.

Питание микросхем +5В±5 %

Арифметическое устройство КР588ВС2

Разрядность данных 16
 Разрядность микрокоманды 12
 Число каналов ввода-вывода 1
 Максимальный объем адресуемой памяти, К байт 64
 Время цикла, мкс, не более 1,82
 Потребляемая мощность, мВт, не более 4,7

Управляющая память КР588ВУ2

Разрядность команды 16
 Разрядность шины состояний 4
 Разрядность микрокоманды 13
 Число логических произведений 150
 Время выборки микрокоманды, мкс, не более 0,65
 Потребляемая мощность, мВт, не более 16

Системный контроллер КР588ВГ1

Разрядность микрокоманды 5
 Число микрокоманд 26
 Число запросов прерываний 4
 Разрядность кода прерываний 4
 Время цикла, нс 200
 Потребляемая мощность, мВт, не более 4,0

**Многорежимный буферный регистр
 КР588ИР1**

Разрядность 8
 Время записи информации, нс, не более 150
 Время считывания информации, нс, не более 120
 Потребляемая мощность, мВт, не более 0,5

**Магистральный приемопередатчик
 КР588ВА1**

Число приемопередатчиков 8
 Время передачи информации, нс, не более 100
 Потребляемая мощность, мВт, не более 0,5

Контроллер ЗУ КР588ВГ2

Число управляемых модулей ОЗУ 8
 Время цикла, нс 200
 Потребляемая мощность, мВт, не более 0,08

МПК БИС серии К1800 выполнен по биполярной технологии на основе логических элементов ЭСЛ. В этих элементах насыщенные либо отсутству-

ет, либо ограничивается диодами Шотки, благодаря чему обеспечивается наименьшее быстродействие (1,5... 2,5 нс на элемент). Комплект серии К1800 выполнен в виде микропроцессорных секций с возможностью наращивания разрядности элементарным параллельным каскадированием системы. Микропрограммное управление позволяет ориентировать МПК на различные системы команд. Функциональный состав МПК достаточно полный, а непосредственная совместимость со схемами средней степени интеграции и запоминающими устройствами, выполненными на ЭСЛ элементах (серии К500, К100), позволяет значительно расширить возможности комплекта и создавать на его основе быстродействующие ЭВМ, измерительные системы и разнообразные микропроцессорные системы, предназначенные для цифровой обработки информации в реальном масштабе времени.

С помощью двунаправленного транслятора уровня — микросхемы К1800ВА4 — обеспечивается связь комплекта с микросхемами, выполненными по биполярной (ТТЛШ) или МОП-технологии. Микропроцессорные секции комплекта имеют развитую шинную организацию, что упрощает принципы их взаимодействия с другими схемами и позволяет обрабатывать данные, представленные в двоичном и двоично-десятичном кодах в режимах с фиксированной и плавающей запятой.

При использовании МПК серия К1800 необходимо учитывать, что микросхемы имеют значительную потребляемую мощность, и предусматривать меры для снижения температуры корпуса микросхем.

Микросхемы выполнены в прямоугольных металлокерамических корпусах:

- 2207.48-1 — К1800ВС1, К1800ВТ3, К1800ВУ1, К1800ВР8, К1800РП6;
- 2120.24-1 — К1800ВБ2;
- 2103.16-3 — К1800ВА4, К1800ВА7.

Напряжения питания:

$U_{cc} = -5,2 В \pm 5 \%$ для ИС К1800ВБ2, К1800ВР8, К1800РП6, К1800ВА7;
 $U_{cc} = -5,2 В \pm 5 \%$; $U_{cc2} = -2,0 В \pm 5 \%$ — для К1800ВС1, К1800ВТ3, К1800ВУ1;
 $U_{cc1} = -5,2 В \pm 5 \%$; $U_{cc3} = 5 В \pm 5 \%$ — для К1800ВТ3.

Для питания выходных каскадов используется источник опорного напряжения $-2,0 В \pm 1 \%$.

МП секция параллельной обработки информации К1800ВС1

Число каналов ввода 1
 Число каналов ввода-вывода 2
 Число разрядов каналов ввода, ввода-вывода 4
 Число управляющих сигналов 17
 Число выполняемых операций 68
 Объем адресуемой памяти, слов 24n

Потребляемая мощность, мВт, не более	1600
Время выполнения операции сложение* (вычитание), нс, не более	41

Устройство синхронизации К1800БЭ2

Число фаз выходных синхросигналов	1...4
Число управляющих сигналов	8
Потребляемая мощность, мВт, не более	735
Частота тактовых сигналов, МГц, не более	36

Схема управления памятью К1800БТЗ

Число каналов ввода	1
Число каналов вывода	1
Число каналов ввода-вывода	3
Число разрядов каналов ввода, вывода и ввода-вывода	4
Объем адресуемой памяти, слов	2 ⁴ⁿ
Число управляющих сигналов	15
Потребляемая мощность, мВт, не более	1700
Время передачи информации без обработки, нс, не более	11

Многоадресный сдвигатель К1800ВР8

Число разрядов каналов входной и выходной информации	16
Число различных типов сдвигов (арифметические, логические, циклические и т. д.)	8
Число разрядов, на которое можно осуществить сдвиг, не более	16
Потребляемая мощность, мВт, не более	1800
Время выполнения сдвига, нс, не более	80

Двухадресный блок регистров К1800РП8

Организация	32 слова × 9 разрядов
Число разрядов адреса каналов А и В	5
Число разрядов каналов выходной информации (АД и ВД)	9
Потребляемая мощность, мВт, не более	1800
Время записи (считывания информации), нс, не более	18

Двухадресный транслятор уровней К1800ВА4

Число разрядов	4
Потребляемая мощность, мВт, не более	700
Время передачи информации ЭСЛ-ТТЛ, нс, не более	8

Двухадресный магистральный транслятор К1800ВА7

Число разрядов	5
Потребляемая мощность, мВт, не более	440
Время передачи информации из канала А в канал В, нс, не более	6

Схема микропрограммного управления К1800ВУ1

Число каналов ввода	1
Число каналов вывода	2
Число каналов ввода-вывода	2
Число разрядов каналов ввода, вывода и ввода-вывода	4
Объем адресуемой памяти	2 ⁴ⁿ
Число разрядов микрокоманды	4
Число управляющих сигналов	9
Потребляемая мощность, мВт, не более	1900
Время выдачи следующего адреса, нс, не более	33

МПК БИС серии К1801 выполнен на основе л-МОП-технологии и является комплектом среднего быстродействия и средней потребляемой мощности. Его основу составляет однокристалльный 16-разрядный МП, имеющий фиксированную систему команд, совместимую с системой команд микроЭВМ «Электроника 60». Число выполняемых команд — 68. Команды «умножение» и «деление» выполняются программным способом. Объем адресуемой памяти — 64К байт.

Микропроцессор осуществляет обработку внешних и внутренних прерываний и организует обмен информацией с внешними устройствами в соответствии с ГОСТ 26765.51-86. Число уровней прерывания — 4.

Интерфейсные схемы МПК серии К1801 выполнены на основе базового кристалла К1801ВП1, который позволяет при минимальных производственных затратах получать микросхемы с разнообразными функциональными возможностями. Схемы, выполненные на основе базового кристалла К1801ВП1, в основном являются заказными и поставляются в соответствии с договором между заказчиком и исполнителем. Часть микросхем К1801ВП1, указанная ниже, являются схемами широкого применения и поставляется потребителям на общих основаниях.

Микросхема К1801ВП1-030 представляет собой схему управления блоком ОЗУ, выполненном на основе микросхем ОЗУ динамического типа (К565РУ6). Микросхема осуществляет прием, хранение и преобразование адреса для накопителя ОЗУ; регенерацию памяти, связь накопителя ОЗУ и буферного регистра данных с каналом передачи информации микроЭВМ типа «Электроника 60».

Микросхемы К1801ВП1-033 и К1801ВП1-034 — многофункциональные устройства, режим работы которых и выполняемая функция задаются управляющими сигналами, подаваемыми на соответствующие выводы.

Микросхема К1801ВП1-033 может работать в режимах интерфейса накопителя на гибких магнитных дисках, контроллера интерфейса параллельного ввода-вывода, контроллера байтового параллельного интерфейса. Микросхема К1801ВП1-034 — в режимах 8-разрядного устройства передачи информации, 16-разрядного буферного регистра данных, устройства выдачи вектора прерывания и компаратора адреса.

Микросхема К1801ВМ1-035 представляет собой асинхронный приемопередатчик для внешних устройств, работающих на линии связи с последовательной передачей информации. Предназначена для преобразования параллельной информации в последовательную и наоборот.

Однокристалльные микропроцессоры К1801ВМ1 А, Б, В

Разрядность обрабатываемых данных	16
Число выполняемых команд	68
Максимальный объем адресуемой памяти, К байт	64
Число каналов передачи информации	1
Число уровней прерывания	4
Виды адресации: регистровая, косвенно-регистровая, автоинкрементная, косвенно-автоинкрементная, индексная, косвенно-индексная	
Быстродействие, тыс. операций/с (типа регистр-регистр)	500
Тактовая частота, МГц, не более	4,7 (А), 3,5 (Б), 2,5 (В)
Потребляемая мощность, мВт, не более	1200

Устройство управления оперативным запоминающим устройством динамического типа К1801ВП1-030

Число разрядов адреса строки	7
Число разрядов адреса столбца	7
Время цикла регенерации памяти, мс	2
Потребляемая мощность, мВт, не более	1000

Многофункциональное устройство К1801ВП1-033

Число режимов работы	3
Потребляемая мощность, мВт, не более	1000

Многофункциональное устройство К1801ВП1-034

Число режимов работы	3
Потребляемая мощность, мВт, не более	1000

Асинхронный приемопередатчик К1801ВМ1-035

Число разрядов принимаемых и выдаваемых данных, бит	5...8
Максимальная скорость передаваемой информации, Бод (при тактовой частоте 4508 кГц)	19 200
Потребляемая мощность, мВт, не более	1000

МПК БИС серии КР1802 выполнен на основе биполярной технологии (ТТЛШ) и предназначен для построения быстродействующих контроллеров периферийных устройств, встроенных и автономных микро- и мини-ЭВМ, устройств автоматики, аппаратных умножителей, устройств для быстрого преобразования Фурье и т.д.

Разнообразная номенклатура, возможность параллельного наращивания разрядности, микропрограммный способ управления, совместимость с ТТЛ и ТТЛШ сериями позволяют широко использовать этот МПК совместно с микросхемами других серий (К589, КМ1804) для создания МСВТ разнообразного назначения.

В состав МПК входит ряд аппаратных умножителей различной разрядности (8×8, 12×12, 16×16), которые могут использоваться как самостоятельно, так и с другими МПК при условии согласования их по входным и выходным уровням. Умножители производят умножение кодов (чисел без знака) и чисел со знаком, представленных в дополнителем кода. Числа могут быть как целыми, так и меньше единицы.

Микросхемы выполнены в пластмассовых (КР) и керамических (КМ) корпусах.

239.24-2	— КР1802ИР1;
2206.42-1	— КР1802ВС1,
	КР1802ВР1,
	КР1802ВВ1;
2206.42-2	— КР1802ВР3;
2207.48	— КР1802ИМ1;
2136.64-1	— КМ1802ВР4,
	КМ1802ВР5.

Питание микросхем +5В±5%

МП секция параллельной обработки информации КР1802ВС1

Разрядность обрабатываемых данных	8
Число каналов ввода-вывода	2
Разрядность каналов ввода-вывода	8
Число разрядов микрокоманды	8
Максимальный объем адресуемой памяти, слов	2 ⁸ⁿ
(n — число используемых микросхем)	
Время передачи информации от входа до выхода, нс, не более	150
Потребляемая мощность, мВт, не более	1400

Двухадресный блок регистров общего назначения КР1802ИР1

Число адресных шин	2
Разрядность адресных шин	4
Число информационных шин	2
Разрядность информационных шин	4
Число регистров	16
Разрядность каждого регистра	4
Время передачи информации с шины DA на шину DB, нс, не более	45
Потребляемая мощность, мВт, не более	800

Арифметический расширитель (сдвигатель) КР1802ВР1

Число каналов ввода-вывода	2
Разрядность канала ввода-вывода	16
Разрядность канала параметра сдвига	5
Число каналов ввода	1
Разрядность канала ввода	5
Разрядность микрокоманды	3
Время передачи информации от входа до выхода информации, нс, не более	90
Потребляемая мощность, мВт, не более	1400

Схема обмена информацией КР1802ВВ1

Число каналов ввода-вывода	4
Разрядность каждого канала ввода-вывода	4
Время передачи информации из канала в канал, нс, не более	80
Потребляемая мощность, мВт, не более	1400

Схема умножителя 8x8 разрядов КР1802ВР3

Число каналов ввода	2
Разрядность каналов ввода	8
Число каналов вывода	1
Разрядность канала вывода	16
Время умножения 8-разрядных чисел, нс, не более	130
Потребляемая мощность, мВт, не более	1350

Схема умножителя 12x12 разрядов КМ1802ВР4

Число каналов ввода	2
Разрядность каналов ввода	12
Число каналов вывода	2
Разрядность каналов вывода	12
Время умножения двенадцатиразрядных чисел, нс, не более	180
Потребляемая мощность, мВт, не более	3000

Схема умножения 16x16 разрядов КМ1802ВР5

Число каналов ввода	1
Разрядность канала ввода	16
Число каналов ввода-вывода	1
Разрядность канала ввода-вывода	16
Число каналов вывода	1

Разрядность канала вывода	16
Время умножения 16-разрядных чисел, нс, не более	165
Потребляемая мощность, мВт, не более	4000

Сумматор четырех 4-разрядных чисел КР1802ИМ1

Число каналов ввода	4
Разрядность каналов ввода	4
Число каналов вывода	1
Разрядность канала вывода	4
Время суммирования, нс, не более	47
Потребляемая мощность, мВт, не более	1400

МПК БИС серии КМ1804 (КР1804) выполнен на основе биполярной технологии (ТТЛШ) и является одним из быстродействующих и функционально полных комплектов. В его состав входит 18 микросхем различного функционального назначения, и проводятся работы по дальнейшему его расширению.

Основой для построения операционных блоков вычислительных устройств является 4-разрядная микропроцессорная секция КМ1804ВС1 (КМ1804ВС2) с возможностью параллельного наращивания разрядности. Микросхема КМ1804ВС2 по сравнению с микросхемой КМ1804ВС1 выполняет большее число арифметическо-логических функций и дополнительно реализует: умножение, деление, нормализацию, преобразование в дополнителный код и т. д., что позволяет существенно повысить быстродействие систем и сэкономить объем требуемой памяти программ. Кроме того, микросхема КМ1804ВС2 обеспечивает возможность расширения файла регистров общего назначения и осуществляет контроль четности.

МПК серии КМ1804 имеет микропрограммный способ управления, реализуемый с помощью микросхем КМ1804ВУ1 (КМ1804ВУ2), и КМ1804ВУ3 и внешнего ПЗУ команд. Для систем с большим набором команд вместо микросхем КМ1804ВУ1 и КМ1804ВУ3 целесообразно использовать микросхему КМ1804ВУ4.

Микропрограммный способ управления позволяет создавать устройства с различными системами команд, однако при этом необходимо учитывать, что оригинальные системы команд требуют самостоятельной разработки программного обеспечения.

Микросхемы серии КМ1804 выполнены в металлокерамических корпусах

201.16-13	— КМ1804ВУ3,
	КМ1804ВР1,
	КМ1804ИР1;
2140Ю.20-2	— КМ1804ВУ2,
	КМ1804ВА2,
	КМ1804ВР3;
2108.22-1	— КМ1804ИР2;
2120.24-1	— КМ1804ВА1,
	КМ1804ВА3;
	КМ1804ГГ1;
2121.28-6	— КМ1804ВУ1,
	КМ1804ИР3,
	КМ1804ВУ5;

2123.40-6	— КМ1804ВС1,
	КМ1804ВР2,
	КМ1804ВУ4,
	КМ1804ВН1;
2126.48-1	— КМ1804ВС2,
	КМ1804ВЖ1.

Значительная часть МПК серии К1804 выпускается в пластмассовых корпусах. Такие микросхемы имеют обозначение КР1804. Их функциональное назначение и параметры соответствуют аналогичным по наименованию типоминалам МПК серии КМ1804.

Микросхемы серии КР1804 выполнены в корпусах

201.16-16	— КР1804ВУ3,
	КР1804ВР1,
	КР1804ИР1,
2140Ю.20-1	— КР1804ВУ2,
	КР1804ВА2,
	КР1804ВР3;
239.24-7	— КР1804ВА1,
	КР1804ВА3,
	КР1804ГГ1;
2121.28-4	— КР1804ВУ1;
2123.40-1	— КР1804ВС1,
	КР1904ВР2,
	КР1804ВУ4,
	КР1804ВН1

Питание микросхем +5В±5%

Микропроцессорная секция параллельной обработки информации КМ1804ВС1

Число каналов ввода	3
Число каналов вывода	1
Разрядность каналов	4
Число регистров общего назначения	16
Разрядность регистра общего назначения	4
Разрядность микрокоманды	9
Объем адресуемой памяти, слов	2 ⁴ⁿ
Длительность цикла тактовых сигналов, нс, не более	100
Потребляемая мощность, мВт, не более	1470

Микропроцессорная секция параллельной обработки информации КМ1804ВС2

Число каналов ввода	3
Число каналов ввода-вывода	2
Разрядность каналов	4
Число регистров общего назначения	16
Разрядность регистров общего назначения	4
Разрядность микрокоманды	9
Объем адресуемой памяти, слов	2 ⁴ⁿ
Длительность цикла тактовых сигналов, нс, не более	104
Потребляемая мощность, мВт, не более	1837

Схемы управления адресом микрокоманды КМ1804ВУ1/КМ1804ВУ2

Число каналов ввода	3
КМ1804ВУ1	1
КМ1804ВУ2	1
Число каналов вывода	1
Разрядность каналов	4
Объем адресуемой памяти, слов	2 ⁴ⁿ
Время передачи информации от входа тактового сигнала до выхода, нс, не более	102
Потребляемая мощность, мВт, не более	683

Схема управления следующим адресом КМ1804ВУ3

Число формируемых управляющих микрокоманд	16
Число входов	5
Число выходов	8
Время передачи информации от входа до выхода, нс, не более	60

Потребляемая мощность, мВт, не более 604

Схема управления последовательностью микрокоманд КМ1804ВУ4
 Объем адресуемой памяти, слов 4096
 Разрядность адреса 12
 Глубина стека 5
 Число выполняемых микроинструкций 16
 Время передачи информации от входа тактового сигнала до выхода, нс, не более 125
 Потребляемая мощность, мВт, не более 1806

Параллельный регистр КМ1804ИР1
 Число каналов ввода 1
 Число каналов вывода 2
 Разрядность каналов 4
 Время передачи информации от входа тактового сигнала до выхода, нс, не более 21
 Потребляемая мощность, мВт, не более 683

Схема ускоренного переноса КМ1804ЕР1
 Число разрядов 4
 Время передачи информации от входа до выхода, нс, не более 19
 Потребляемая мощность, мВт, не более 572

Схема управления состоянием и сдвигами КМ1804ВР2
 Число каналов ввода 2
 Число каналов ввода-вывода информации 1
 Разрядность каналов 4
 Разрядность микрокоманды 13
 Число выполняемых типов сдвигов 32
 Время передачи информации от входа тактового сигнала до выхода условия, нс, не более 58
 Потребляемая мощность, мВт, не более 1670

Магистральный приемопередатчик КМ1804ЕА1
 Число каналов ввода 2
 Число каналов вывода 1
 Число каналов ввода-вывода информации 1
 Разрядность каналов 4
 Время передачи информации от входа до выхода, нс, не более 42
 Потребляемая мощность, мВт, не более 525

Магистральный приемопередатчик КМ1804ВА2
 Число каналов ввода 1
 Число каналов вывода 1
 Число каналов ввода-вывода 1
 Разрядность каналов 4
 Контроль четности Есть
 Время передачи информации от входа до выхода, нс, не более 44
 Потребляемая мощность, мВт, не более 600

Магистральный приемопередатчик КМ1804ВА3
 Число каналов ввода 2
 Число каналов вывода 1
 Число каналов ввода-вывода информации 1
 Разрядность каналов 4
 Контроль четности Есть
 Время передачи информации от входа до выхода, нс, не более 50
 Потребляемая мощность, мВт, не более 550

Параллельный регистр КМ1804ИР2
 Число разрядов 8
 Время передачи информации от входа тактового сигнала до выхода, нс, не более 45
 Потребляемая мощность, мВт, не более 185

Параллельный двунаправленный регистр КМ1804ИР3
 Число каналов ввода-вывода 2
 Разрядность каналов ввода-вывода 8

Время передачи информации от входа тактового сигнала до выхода, нс, не более 26
 Потребляемая мощность, мВт, не более 1375

Генератор тактовых импульсов КМ1804ГГ1
 Число выходных фаз 4
 Число режимов работы 4
 Частота опорного генератора, МГц, не более 30
 Потребляемая мощность, мВт, не более 600

Схема векторного приоритетного прерывания КМ1804ВН1
 Число запросов прерывания 8
 Разрядность вектора прерывания 3
 Число разрядов микрокоманды 4
 Время передачи информации от входа тактового сигнала до выхода сигнала «прерывание», нс, не более 97
 Потребляемая мощность, мВт, не более 1525

Расширитель приоритетного прерывания КМ1804ВР3
 Число входов 8
 Число выходов 3
 Время передачи информации от входа до выхода, нс, не более 48
 Потребляемая мощность, мВт, не более 120

Секция управления адресом программной памяти КМ1804ВУ5
 Число каналов ввода 1
 Число каналов вывода 1
 Разрядность каналов 4
 Число разрядов микрокоманды 5
 Время передачи информации от входа до выхода, нс, не более 69
 Потребляемая мощность, мВт, не более 1100

Схема обнаружения и коррекции ошибок КМ1804ВЖ1
 Число каналов ввода-вывода 2
 Разрядность каналов ввода-вывода 8
 Разрядность входной и выходной шины контрольных битов 7
 Время передачи информации от входа до выхода, нс, не более 61
 Потребляемая мощность, мВт, не более 2000

МПК БИС серии КМ1810 (КР1810) на уровне языка ассемблера программно совместим с МПК серии КР580. МП серии КМ1810 является 16-разрядным, имеет развитую систему команд (135 команд), обладает высоким быстродействием и отличается рядом функциональных особенностей, позволяющих строить на его основе мощные высокопроизводительные мультипроцессорные системы. Высокая производительность собственно МП серии КМ1810 (КМ1810ВМ86) достигается благодаря совмещению выполнения операций обработки и обращения, что обеспечивается путем предварительной выборки и хранения команд. Отличительной особенностью МПК серии КМ1810 является возможность работы в двух режимах: минимальном и максимальном. Минимальный режим используется в системах, имеющих несложную конфигурацию. При этом МП сам вырабатывает все необходимые сигналы управления периферийными устройствами.

Максимальный режим применяется при использовании МП в составе сис-

темы сложной конфигурации. В этом случае используется специальная микросхема — контроллер шины (КР1810ВГ88), которая анализирует сигналы состояния МП и в зависимости от их значения формирует соответствующие сигналы управления периферийными устройствами.

При построении многопроцессорной системы на базе МП КМ1810ВМ86 в качестве устройства, обеспечивающего синхронизацию доступа множества ведущих устройств к системной шине, используется арбитр системной шины — микросхема КР1810ВБ89. Арбитраж осуществляется на основе принципа приоритета, т. е. в любой данный промежуток времени одно ведущее устройство будет иметь приоритет над всеми другими устройствами.

Микропроцессор КМ1810ВМ86 выполнен по НМОП-технологии, контроллер прерываний (микросхема КР1810ВН59А) — по п-МОП-технологии, остальные микросхемы комплекта по биполярной (ТТЛШ).

Микросхемы выполнены в металл-керамических и пластмассовых корпусах:
 2104.18-5 — КР1810ГФ84: 2140Ю.20-1 — КР1810ВГ88, КР1810ВБ89; 2121.28-5 — КР1810ВН59А; 2123.40-6 — КМ1810ВМ86.

Центральный процессор КМ1810ВМ86
 Разрядность арифметического устройства 16
 Шины адреса и данных совмещены и имеют разрядность 20/16
 Объем адресуемой памяти, Мбайт 1
 Число базовых команд 135
 Число адресуемых устройств ввода-вывода 216/216
 Число режимов адресации 24
 Число внутренних 16-разрядных регистров:
 общего назначения 4
 индексных 2
 указателей 2
 сегментных 4
 Число внешних запросов прерывания 2
 Число внутренних программных запросов прерывания 3
 Число уровней прерывания 256
 Частота тактовых сигналов, МГц, не более 5
 Время выполнения команд типа регистр-регистр, мкс пересылка 0,4
 сложение 0,6
 умножение 23,6...26,6
 деление 28,6...32,4
 Потребляемая мощность, мВт, не более 1800

Генератор тактовых импульсов КР1810ГФ84
 Максимальная частота входного тактового сигнала или опорного генератора, МГц, не более 25
 Частота выходных тактовых п-МОП сигналов F/3
 Частота выходных тактовых ТТЛ-сигналов F/6
 Потребляемая мощность, мВт, не более 700

Контроллер шины КР1810ВГ88
 Число разрядов шины состояния 3
 Частота тактовых сигналов, МГц, не более 10
 Потребляемая мощность, мВт, не более 1150

В XII пятилетке в нашей стране планируется выпуск около 1 млн. персональных компьютеров, близких по своей архитектуре и периферийному оборудованию к РС XT фирмы ИБМ или ее дальнейшим модификациям. Поэтому редакция литературы по информатике и робототехнике издательства «Мир», начиная с 1985 г., приступила к выпуску книг по персональным компьютерам. В 1986—1987 гг. будут выпущены в свет книги по использованию персональных компьютеров в расчетно-теоретических работах, выпуске чертежно-конструкторской документации, использованию их в административно-хозяйственной деятельности предприятий, автоматизации подготовки технологических карт и управления производственными процессами.

Редакция считает, что эти книги должны быть рассчитаны на три категории читателей: первые — специалисты, разрабатывающие персональные компьютеры, создающие для них операционные системы, базы данных и специализированное программное обеспечение; вторые — инженеры, обладающие знаниями в области программирования и использующие персональные ЭВМ в решении своих практических задач; и, наконец, третьи — неспециалисты — пользователи, начинающие осваивать основы вычислительной техники и программирования. Естественно, что для каждой категории читателей должны быть предусмотрены к изданию свои типы книг.

Предлагаемая вниманию читателей книга Р. Трейстера «Персональный компьютер фирмы ИБМ» рассчитана на начинающих пользователей и написана неспециалистом в области вычислительной техники и программирования. В процессе освоения персонального компьютера фирмы ИБМ автору книги пришлось изложить вопросы, связанные с установкой ЭВМ, ее эксплуатацией, расширением вычислительных возможностей, обнаружением неисправностей, профилактическими работами и программированием. В книге даны несколько простых примеров и расширенное приложение, позволяющие упростить освоение работы на персональном компьютере фирмы ИБМ.

Автор книги не всегда придерживался строгой терминологии в изложении операторов языка Бейсик. Например, в описании операторов LINE INPUT, LET, CHAIN, WHILE и некоторых других (гл. 7) рецензенты Ю. Л. Кетков и Г. В. Сенин* рекомендуют пользоваться принятой в нашей стране терминологией или переводчику книги дать соответствующие примечания, так как такая коррекция, безусловно, улучшила бы содержание 7-й главы книги.

Нам хотелось бы ознакомить читателей журнала «Микропроцессорные средства и системы» с отдельными высказываниями читателей данной книги, которые были получены редакцией литературы по информатике и робототехнике. Как правило, эти письма составлены читателями, имеющими доступ к персональным компьютерам фирмы ИБМ. Вот краткие выдержки из этих писем:

«По сути дела в этой книге имеются все необходи-

мые материалы для освоения компьютера начинающим пользователем. Нужно издательству больше выпускать книг, посвященных персональным компьютерам. Хотелось бы в заключение еще раз поблагодарить издательство «Мир» за выпуск крайне полезной и нужной многим читателям книги. О том, что данная книга нашла своих читателей, говорит тот факт, что приобрести ее в книжных магазинах не представляется возможным».

«Очень интересна та часть книги, где описывается схема гарантийного и профилактического обслуживания персонального компьютера. Считаю, что этот материал мог бы послужить темой для широкой дискуссии и заслуживает опубликования в центральной прессе, газетах «Известия», «Литературная газета» и т. д. Это очень интересная, нужная и своевременная у нас книга».

«Русский текст книги читается легко, и все ее материалы понятны неподготовленному читателю. К недостатку книги следует отнести смехотворно малый ее тираж. По моему мнению и мнению моих коллег, книга разошлась бы мгновенно при тираже в 5 раз больше нынешнего».

«Просим издательство сообщить, где можно приобрести данную книгу. Необходимо выпустить ее 2-е издание».

Преподаватели инженерного вуза отмечают, «...что они с большим удовлетворением встретили выпуск издательством «Мир» книги Р. Трейстера «Персональный компьютер фирмы ИБМ». Книга написана простым и ясным языком, не требует для ее освоения никаких предварительных знаний в области вычислительной техники. Вызывает большое удивление крайне малый тираж книги».

«Пользователи нуждаются в хороших книгах по аппаратному и программному обеспечению персональных ЭВМ. К числу их можно отнести книгу Р. Трейстера «Персональный компьютер фирмы ИБМ». Хотя в ней не везде выдержан хороший стиль перевода и имеются отдельные неточности в описании операторов языка Бейсик, тем не менее простота изложения фрагментов программ и доступность их для читателей не нуждаются ни в каких дополнительных пояснениях».

Некоторые авторы писем указывают на отдельные неточности в изложении. «В книге «Персональный компьютер фирмы ИБМ» имеются в двух программах две ошибки. Хотелось, чтобы их не было, так как на отладку этих программ мне пришлось затратить 3 часа». Авторы писем указывают и на неточности сопоставления операторов WHILE—WEND с оператором GOSUB—RETURN и некоторые другие опечатки в оригинале.

Поэтому можно утверждать, что книга Р. Трейстера нашла своего читателя и пользуется у него большим спросом. Об этом свидетельствуют многочисленные письма, получаемые редакцией.

Нам хотелось бы отметить, что редакция литературы по информатике и робототехнике пока не располагает персональным компьютером и поэтому не может обеспечить проверку правильности составленных автором программ. То, что эта работа требует больших затрат времени, можно проиллюстрировать на примере пере-

Арбитр шины KP1810VB689	
Число разрядов шины состояния	3
Число режимов работы	4
Число методов разрешения приоритета	3
Частота тактовых сигналов, МГц, не более	10
Потребляемая мощность, мВт, не более	865

Программируемый контроллер прерываний KP1810VH59A	
Число обслуживаемых запросов прерывания	8
Разрядность шины данных	8
Число программируемых режимов работы	3
Время выдачи сигнала «прерыва-	

ние» на сигнал «запрос», нс, не более	350
Потребляемая мощность, мВт, не более	450

Телефон для справок: 536-57-55, Москва

Статья поступила 5 мая 1987 г.

вода книги Уилкинсона и Райнша «Справочник алгоритмов на языке Алгол» (М.: Машиностроение, 1976), переводчики которой провели проверку 68 программ из 84 на БЭСМ-4 и БЭСМ-6. На это было затрачено около 2-х лет работы трех программистов. Тем не менее в непроверенных программах Сибирским отделением АН

СССР было обнаружено 12 ошибок в оригинале и наборе.

В заключение редакция литературы по информатике и робототехнике выражает признательность Ю. Л. Кеткову и Г. В. Сенину за сделанные замечания по гл. 7 «Бейсик персонального компьютера фирмы ИБМ».

ОТВЕТ

переводчика
на критические замечания
Ю. Л. Кеткова и Г. В. Сенина
на книгу Р. Трейстера
«Персональный компьютер
фирмы ИБМ»

В начале ответа следует отметить, что оба оппонента являются высококвалифицированными специалистами в области программирования на Бейсике. Ю. Л. Кетков — автор известной книги, посвященной программированию на Бейсике, а Г. В. Сенин, судя по его отзыву, имеет большой опыт работы на персональном компьютере фирмы ИБМ (ПК ИБМ). Поэтому их мнение о книге квалифицированное, и к нему, безусловно, следует прислушаться. Все их замечания к переводу текста справедливы, хотя я и не везде согласен с их трактовкой. Но в любом случае книга выиграла бы, если бы удалось устранить эти недочеты, что, к сожалению, уже невозможно сделать.

Я писал в «Предисловии переводчика» и хочу повторить еще раз, что данная книга не является ни монографией, ни учебным пособием. Она предназначена для предварительного общего знакомства с ПК. Основное в данной книге — не детали программирования и особенности действия тех или иных операторов. Главная цель книги — дать широкому кругу читателей, не обладающих специальными знаниями в области ЭВМ (учителям, бухгалтерам, администраторам, школьникам и т. д.), общее представление о том, что такое ПК, какой уровень сложности этого устройства, что нужно знать и уметь, чтобы им можно было пользоваться не только в своей практической деятельности, но и быту; какими средствами достигаются удивительные качества, которыми располагает ПК. На достоинства и недостатки этой книги и ее перевода нужно смотреть с позиций указанной категории читателей, а не с узкопрофессиональной точки зрения.

Оба оппонента придерживаются естественной для специалистов точки зрения, согласно которой программирование рассматривается, как часть математической науки со всеми вытекающими отсюда последствиями в отношении строгости терминологии и форматирования. Автор книги придерживается другой точки зрения, а

именно, что программирование — ремесло, которым нужно овладеть в той степени, которая требуется для конкретно выполняемой работы, а тонкости программирования следует постигать в тот момент, когда возникают трудности, а не с самого начала. Отсюда следует нестрогий стиль изложения (с возможными неточностями), но зато большим выигрышем является доступность изложения, что важно для неквалифицированного читателя. Я, как переводчик, обязан был сохранить этот стиль.

Далее я хотел бы ответить на критические замечания по поводу качества перевода. Я буду придерживаться, в основном, отзыва Г. В. Сенина, так как оба отзыва содержат в принципе одни и те же замечания.

По мнению Г. В. Сенина, я должен был прокомментировать два неправильных, по его мнению, утверждения автора:

1. Сопоставление операторов WHILE—WEND с операторами GOSUB—RETURN поверхностно, примитивно... неуместно. С точки зрения теории программирования с такой оценкой оппонента можно согласиться, однако мысль автора, тем не менее, нельзя считать совершенно неправильной. Во всяком случае, можно представить себе отдельные частные случаи, когда эти пары операторов работают аналогично.

2. «Автор неправильно сопоставляет операторы GOTO и GOSUB». Такое сопоставление неправомерно и здесь я полностью согласен с замечаниями отзыва.

Далее отвечу на последующие замечания по поводу перевода:

1. Об операторе AUTO. Я считаю, что перевод адекватен оригиналу и правильно отражает существо дела. Возражение оппонентов вызывает фраза «ссылаться на номер предшествующей строки». Однако если бы понадобилось написать программу, описывающую процедуру вычисления номера строки, и копирующую действия программиста, то она обязательно содержала бы ссылку с целью обращения к предыдущей строке.

2. Об операторе LET. Замечание, касающееся неточности в определении самого оператора, справедливо. Вопрос о том, как трактовать отсутствие слова LET, кажется мне чисто теоретическим и не представляющим интереса для программирования.

3. К сожалению, автор книги подменяет один механизм языка (пересылка значений) другим (использование общих областей памяти). Это относится к оператору COMMON.

С другими замечаниями, касающимися собственно перевода и опечаток, я согласен.

Оценивая критические замечания в целом, я считаю, что большинство из них носит не принципиальный, а чисто технический характер. Большая часть замечаний связана с тем, что многие слова, соответствующие вполне определенным понятиям в теории программирования (ссылка, пересылка, условный и т. п.) употреблялись автором и мною в отличающемся от этих понятий, но все же правильном принятом в обычном языке смысле. Учитывая, что цель книги — первоначальное знакомство с персональным компьютером лиц, не имеющих никакой подготовки, я думаю, что недостатки, указанные оппонентами, не являются определяющими для широкого круга читателей. Они могут вызвать чувство досады лишь у профессионалов-специалистов высокого класса, таких, как мои оппоненты, которым эта книга абсолютно не нужна.

В заключение я хотел бы прокомментировать оценку, данную книге Г. В. Сениным, вне ее связи с качеством перевода.

Г. В. Сенин считает, что выбор книги для перевода сделан неудачно. Я не согласен с его мнением. В настоящее время основная проблема в освоении ЭВМ — наличие психологического барьера у людей, особенно среднего и старшего поколения. Существует очень много заблуждений и предрассудков относительно возможностей ЭВМ и средств ими предоставляемых. Поэтому очень нужна литература «для психологов», где можно было бы, не имея специальных знаний, ознакомиться с основами ЭВМ и их возможностями и подготовить себя к изучению более серьезной литературы. Книга Р. Трейстера преследует именно эту цель, и, на мой взгляд, автор ее достиг.

Я хочу поблагодарить Ю. Л. Кеткова и Г. В. Сенина за их участие в обсуждении книги, свидетельствующее об их заинтересованности в скорейшем развитии вычислительной техники у нас в стране.

С уважением
Н. Г. Волков

РЕФЕРАТЫ СТАТЕЙ

УДК 681.327.11

Жуков Н. Н., Коризенков А. С., Офицеров В. Н. Печатающее устройство «Электроника МС6307» // «Микропроцессорные средства и системы». — 1987. — № 4. — С. 3.

Рассмотрено печатающее устройство, предназначенное для работы в составе вычислительных комплексов с персональными ЭВМ «Электроника МС0585» или другими, имеющими аналогичный интерфейс и протокол обмена.

UDC 681.327.11

Zhukov N. N., Korisenkov A. S., Ofitserov V. N. "Electronica MC6307" printer. // "Microprocessor devices and systems". — 1987. № 4. — P. 3.

"Electronica MC6307" printer, designed for use in computer systems with "Electronica MC0585" personal computer, is introduced. The printer may be used also with other computers having compatible printer interface and data transfer protocol.

УДК 681.325.5:681.327.8.06

Габдуллин Р. Р., Горемыкин В. В., Горяшко А. П., Косов Л. С., Лукович Г. А., Мионов В. Г. Комплект БИС для встроенного диагностирования микропроцессорных систем // Микропроцессорные средства и системы. — 1987. — № 4. — С. 4.

Описан комплект БИС, предназначенный для аппаратной реализации методов тестового диагностирования микропроцессорных систем. Рассмотрены общие принципы встроенного диагностирования, режимы работы и способы применения комплекта в микропроцессорных системах.

UDC 681.325.5:681.327.8.06

Gabdullin R. R., Goremykin V. V., Goryashko A. P., Kosov L. S., Lukovich G. A., Mironov V. G. LSI set for built-in diagnostics in microprocessor systems. // "Microprocessor devices and systems". — 1987. — N 4. — P. 4.

The article gives a description of LSI chips, which may be used for hardware implementation of some diagnostic methods in microprocessor systems. General principles of built-in diagnostics, LSI set operation modes and its applications in microprocessor devices are explained.

УДК 681.321 — 181.48

Лемко Л. М., Гладков В. В., Ермаков С. В., Жуков В. Н. Персональный микрокомпьютер «Электроника МК-85» // «Микропроцессорные средства и системы», — 1987. — № 4. — С. 13.

Описан карманный персональный компьютер «Электроника МК-85», работающий в режимах обычного микрокалькулятора и ЭВМ с использованием версии языка Бейсик.

UDC 681.321—181.48

Lemko L. M., Gladkov V. V., Ermakov S. V., Zhukov V. N. Personal microcomputer "Electronika MK-85". // "Microprocessor devices and systems". — 1987. — N 4. — P. 13.

A pocket personal microcomputer "Electronika MK-85" which can operate in both calculator and BASIC-programmable computer modes is featured.

УДК 658.012.011.56

Казменко С. В. Автоматизация проектирования систем на основе объединения концепций макетирования и «программирования без программиста» // Микропроцессорные средства и системы. — 1987. — № 4. — С. 16.

Предлагается создать такие инструментальные средства, которые позволяли бы разрабатывать прикладные системы без привлечения программистов. Рассматривается возможность автоматизации создания, таким образом, баз данных, средств взаимодействия пользователя с ЭВМ и средств получения машинограмм.

UDC 658.012.011.56

Kazmenko S. V. System design automation based on united concepts of prototyping and "programming without programmer" // "Microprocessor devices and systems". — 1987. — N 4. — P. 16.

The author proposes to create "system construction kits", the software tools which would enable generation of applied systems without programmers aid. The possibility of application of such method for building database support programs, interactive user interface software and printout utilities is examined.

УДК 681.3.068

Григорьев А. Г. Диалоговый дизассемблер для загрузочных модулей операционной системы РАФОС // «Микропроцессорные средства и системы». — 1987. — № 4. — С. 19.

Рассмотрены возможности программы-дизассемблера, позволяющей получить текст программы на макроасемблере из загрузочного модуля. Генерируемый текст отличается высоким качеством и пригоден для модификации с последующим повторным ассемблированием. Дизассемблер был применен для разработки операционной системы АДОС.

UDC 681.3.068

Grigoryev A. G. Dialogue disassembler utility for .SAV programs running under RAFOS operating system. // "Microprocessor devices and systems". — 1987. — N 4. — P. 19.

The main features of smart dialogue disassembler which can generate MACRO-11 text out of .SAV program modules are illustrated. The quality of generated source text enables its direct editing and re-assembly. Disassembler was used for ADOS operating system design.

УДК 681.326—681.3.06

Воржев А. В., Зверков Б. С., Кикоть А. И., Яковлева Е. В. Интегрированная система разработки расширяемых кросс-систем на ДВК-2М для микропроцессоров. // «Микропроцессорные средства и системы». — 1987. — № 4. — С. 30.

Рассмотрен кросс-транслятор и комплекс технологических средств для проектирования и отладки ПО МПС. Приведены характеристики конкретной кросс-системы RASA-84 для МК БИС КР580.

UDC 681.326—681.3.06

Vorzhev A. V., Zverkov B. S., Kikot A. I., Yakovleva E. V. Integrated expandable cross-systems for microprocessors on DVK-2M. // "Microprocessor devices and systems". — 1987. — N 4. — P. 30.

Cross-assemblers and technological tools for software design and debugging for various microprocessors are discussed. Main features of RASA-84 cross-system for КР580 microprocessor family are shown.

УДК 681.322.1—181.4

Сарьян В. К., Смолич Г. Г. Взаимодействие больших и малых баз данных в автоматизированной системе оформления заявок, ориентированной на индивидуальных пользователей // Микропроцессорные средства и системы. — 1977. — № 4. — С. 37.

Даны аппаратные и программные характеристики автоматизированной системы подготовки и оформления первичных документов по заявкам и их экспертизы, которая сокращает в активном диалоге изобретатель—ЭВМ трудоемкость оформления документов на 30...50 %.

UDC 681.322.1—181.4

Saryan V. K., Smolich G. G. The interaction of small and large data bases in the computerized patent service system, designed for individual users. // "Microprocessor devices and systems". — 1977 — N 4. — P. 37.

The description of hardware and software features of the computerized system for preparation, registration and expertise of primary documents for inventions and patent pending. The system reduces time needed for official registration of documents by 30-50 %. The enhancement is achieved by using direct dialogue "inventorcomputer".

УДК 681.317

Семенов П. А., Процак А. М., Егоров В. П. Цветная графика в микроЭВМ «Электроника 60» и «Электроника НЦ-80» // Микропроцессорные средства и системы. — 1987. — № 4. — С. 47.

Для микроЭВМ «Электроника 60» и «Электроника НЦ-80» описаны аппаратные и программные средства построения контроллеров цветной графики на базе 16-цветного контроллера с разрешением 256×256 элементов изображения. Графическая библиотека программ написана на языках Паскаль и ассемблер.

UDC 681.317

Semenov P. A., Protsak A. M., Egorov V. P. Colour graphics in "Electronika-60" and "Electronika NC-80" microcomputers. // "Microprocessor devices and systems". — 1987. — N 4. — P. 47.

The description of hardware and software of 16-colour graphics videocontroller with 256×256 pixel resolution is given. The graphics support library is written in Assembly and Pascal programming languages.

УДК 681.3.06

Алешин А. Н., Крюков С. Н. Программное обеспечение символично-графического цветного телевизионного дисплея // Микропроцессорные средства и системы. — 1987. — № 4. — С. 56.

Рассматривается пакет программ на языке ассемблер микроЭВМ «Электроника 60», реализующий функции программирования и управления дисплейным модулем «Электроника С5-2106» в символьном и графическом режимах его работы совместно с бытовым цветным телевизором. Приведены фрагменты подпрограмм, входящих в состав пакета. Представлена схема связи дисплейного модуля с микроЭВМ и телевизором.

UDC 681.3.06

Aleshin A. N., Kryukov S. N. Symbol and graphics colour TV-display software support. // "Microprocessor devices and systems". — 1987. — N 4. — P. 56.

The article describes assembly-language software package for "Electronika-60" microcomputer, which supports both alphanumerical and graphics modes of "Electronika-C5-2106" videocontroller, used to drive home colour TV. The fragments of some programs of the package are included for illustration. The connection of videocontroller to TV set and computer is explained.

УДК 681.3.022

Бабкин П. А., Солоненко О. П., Тарасов Б. В., Федорин В. Г. Средства цветной машинной графики для микроЭВМ «Электроника 60» // Микропроцессорные средства и системы. — 1987. — № 4. — С. 60.

Приводится краткое описание комплекса технических и программных средств, реализованных для работы в составе микроЭВМ, аналогичных широко распространенной «Электроника 60», для управления цветным графическим монитором. В качестве устройства управления монитором служит контроллер «Электроника МС 4702». Программные средства цветной машинной графики реализованы на языке макроассемблера MACRO-11 в операционной системе РАФОС и включают в себя два пакета подпрограмм, доступных при написании программ на Фортране.

UDC 681.3.022

Babkin P. A., Solonenko O. P., Tarasov B. V., Fedorin V. G. Colour graphics display hardware and software for "Electronika-60" microcomputer. // "Microprocessor devices and systems". — 1987. — N 4. — P. 60.

The article contains brief description of the plug-in colour graphics system for Q-bus microcomputers like "Electronika-60" and its software support kit. Industrial display controller "Electronika MC 4702" is used to drive colour TV monitor. The software for computer graphics system is written in MACRO-11 assembly programming language under RAFOS operating system, and consists of two program units, which may be called from FORT-RAN-IV programs.

УДК 681.327

Дианов А. П., Шелкунов Н. Н. Организация динамической памяти микросистем. // Микропроцессорные средства и системы. — 1987. — № 4. — С. 75.

Рассмотрены принципы реализации подсистем оперативной памяти с 8- и 16-разрядным интерфейсом на базе микросхем динамического ОЗУ серии К565. Изложены вопросы организации контроллеров динамических ОЗУ. Предложены два устройства системной памяти объемом 64 и 128К байт, ориентированные на шину типа И41.

UDC 681.325

Dianov A. P., Shelkunov N. N. Dynamic memory units for microsystems. // "Microprocessor devices and systems". — 1987. — N 4. P. 75.

The principles of circuit design of RAM modules for 8- and 16-bit microsystems using dynamic RAM chips of K565 family are explained. The methods of RAM controllers design are discussed. The circuit diagrams of two practical projects (64K and 128K byte system RAM cards) with Multibus interface are included.

МАШИННАЯ ГРАФИКА

[Окончание. Начало см. на 2 стр. обложки]

На рис. 2 приведен пример изображения, построенного с помощью метода трассирования лучей. Этот метод позволяет создавать высококачественные изображения, но при этом требует весьма значительных вычислительных затрат. Например, на построение изображений, подобных показанному на рис. 2, затрачиваются многие минуты (а то и часы) работы современного мощного компьютера.

В последнее время значительную популярность приобрели методы создания реалистических изображений, построенные на использовании фрактальных поверхностей, т. е. некоторых нерегулярных форм, определяемых на основе стохастической интерполяции. Концепции фронтальной геометрии позволяют описывать многие реальные формы и объекты: горные ландшафты, береговые линии, цветные географические карты, хлопья снега, ветви деревьев и др.

На рис. 3 показана схема построения фрактальной поверхности, похожей на гору. В пространстве берется некоторый исходный треугольник, и над ним выполняется следующая процедура. Находятся средние точки всех сторон треугольника и переносятся (не обязательно в одной плоскости) на расстояние, пропорциональное длине соответствующей стороны. Коэффициент пропорциональности при этом выбирается случайным образом. Соединяя три получившиеся в результате точки, определяем четыре новых треугольника. Затем та же процедура по очереди применяется к каждому из этих четырех треугольников. Теперь образуется 16 треугольников и т. д. Таким образом, из одного

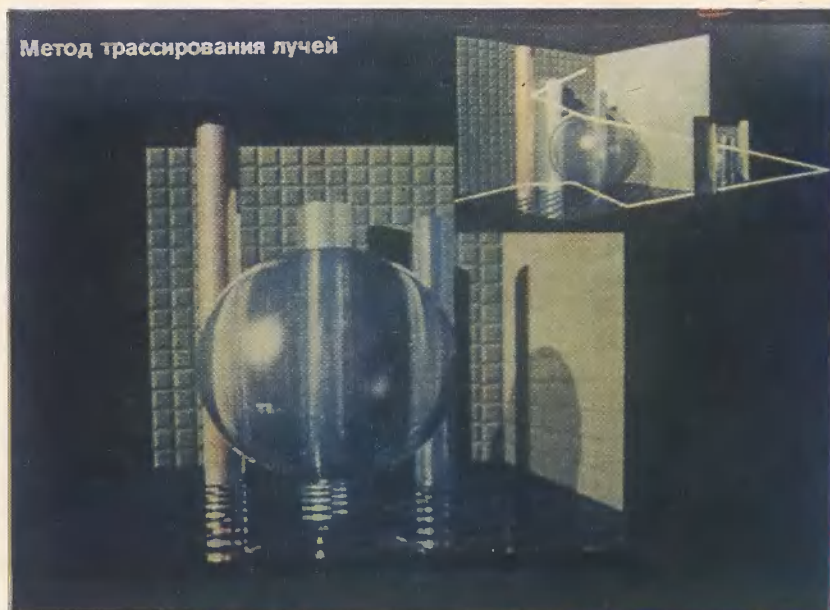


Рис. 2

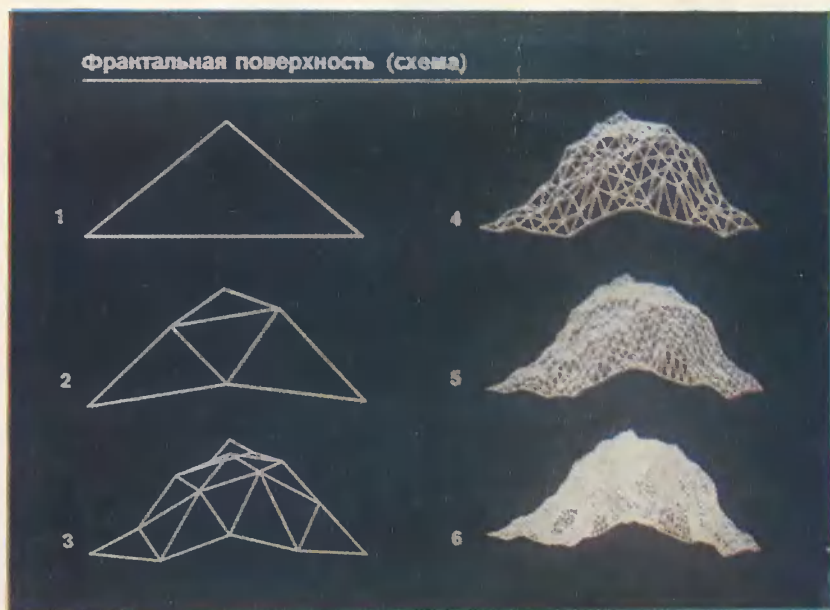


Рис. 3

исходного треугольника можно получить целое множество треугольников, образующих очень сложную полигональную поверхность. Ее изображение создается обычным образом: удаляются скрытые по-

верхности, и затем производится закраска. Заметим, что вместо треугольников в алгоритме могут рассматриваться и четырехугольники, при этом процедура деления лишь незначительно изменится.

ВНИМАНИЕ! ИДЕТ ПОДПИСКА НА 1988 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 первыми узнавать о новинках в области микропроцессорной техники, а также регулярно знакомиться со схемами и примерами программ для самостоятельного создания конкретных устройств автоматизации на базе микроЭВМ и микропроцессоров.

За первые четыре года издания журнал стал настольным, рабочим пособием для профессионалов в области ЭВМ и программирования и вызвал активный практический интерес у самого широкого круга специалистов отраслей народного хозяйства, делающих первые шаги в новый для них мир микропроцессорных средств автоматизации.

Циклы статей из раздела «Учебный центр» помогут Вам на простых примерах практически освоить поистине неисчерпаемые возможности микропроцессорной техники.

На страницах журнала регулярно обсуждаются актуальные проблемы программирования, новейшие тенденции развития информационной технологии. Дискуссии ученых и специалистов по наиболее острым проблемам развития вычислительной техники помогают читателям контролировать «горячие точки» науки и технологии в этой бурно развивающейся области.

Напоминаем, что в розничную продажу журнал не поступает, а достать его в библиотеке, как утверждают читатели, 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 1988 год!

Наш индекс по каталогу «Союзпечать» — 70588. Цена одного комплекта журналов (шесть номеров) — 6 руб. 60 коп.