



МИКРО- ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

1 / 1989

ISSN 0233-4844

ОФОРИС — система подготовки цветных иллюстративных материалов
Асинхронный контроллер динамического ОЗУ предназначен для построения устройств памяти в системах с различными типами МП

Программируемый контроллер прерываний КР1818ВН19 обеспечивает программируемые ответы до 4 байт на каждое прерывание из восьми индивидуально маскируемых приоритетно ранжируемых запросов

Электронный диск в вычислительном комплексе под управлением UNIX-подобной ОС

Программируемый контроллер на базе ОЭВМ КМ1816ВЕ48 позволяет уменьшить число каналов ИРПС в 6 раз и разместить протоколы нижнего уровня управления станками с ЧПУ в памяти ОЭВМ

Работы по созданию законодательства информатики — серьезный шаг по включению экономических факторов, выраженных в юридической форме, в ускорение процесса информатизации в СССР

1	ПЕРО
2	МНОГОУГ
3	КРУГ
4	ЭЛЛИПС
5	ТИП ПЕР
6	РАЗ ПЕР
7	НАП СИМ
8	ФОН СИМ
9	ТРЕКБОЛ
10	УА СЕГМ
11	ПР СЕГМ
12	КОД ПР
13	ИМЕНА
14	МАГ ТР
15	КОНЕЦ



ОФОРИС—ПРИКЛАДНАЯ ГРАФИЧЕСКАЯ СИСТЕМА ФОРМИРОВАНИЯ ИЛЛЮСТРАТИВНЫХ МАТЕРИАЛОВ

(К ст. П. В. Вельтмандера, С. И. Власова)

Система ОФОРИС разработана для подготовки, редактирования, сохранения и показа машинных фильмов. Входной язык — директивного типа.

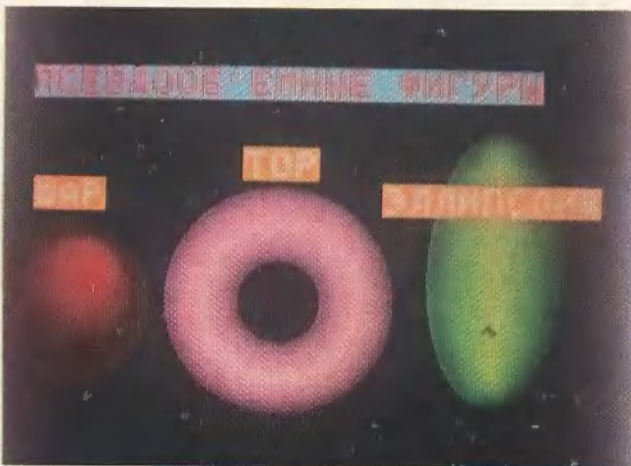
Команда задается либо указанием ее имени в меню координатным устройством, либо вводом числа — номера команды, либо с помощью преддисплейного планшета.

Редактируемое изображение либо его часть берется или из базы данных, или из видеоканала, или создается за пультом.

Результаты работы могут накапливаться в базе данных и документироваться (см. с. 1 обложки).



Плоские примитивные графические элементы системы ОФОРИС: ломаная, строка текста, дуга окружности, закрасненный многоугольник, круг, кольцо, эллипс



Псевдообъемные примитивные графические элементы системы ОФОРИС: шар, тор, эллипсоид с заданием той или иной подсветки



Псевдообъемные фигуры можно нарисовать плоской однотонной кистью или кистью с заданием подсветки следа



Построение изображений копированием с преобразованием плоских примитивов.

Суммарное изображение формируется как из различных примитивов и фигур, так и из результатов геометрических и цветовых преобразований примитивов, уже имеющих на экране. Данное изображение получено копированием кругов с изменением их размеров, расположения и раскраски.

(Продолжение см. на 3 с. обложки)

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года



ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 1/1989 МОСКВА

СОДЕРЖАНИЕ

**МИКРОПРОЦЕССОРНАЯ
ТЕХНИКА**

**Периферийное
оборудование**

**ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ**

**Инструментальные
средства для
отладки программ**

Обработка текстов

Машинная графика

**ПРИМЕНЕНИЕ
МИКРОПРОЦЕССОРНЫХ
СРЕДСТВ**

**Устройства связи
микроЭВМ с объектами
Форум «МП»**

**Справочная
информация**

Артюхов В. Г., Кондратюк В. А., Макеенок А. Н., Пенской В. И., Стяцора Е. В. — Цифровой процессор KM1813BE1: Комплекс средств автоматизации проектирования	2
Бондаренко А. А., Скороходов В. Ф. — Асинхронный контроллер динамического ОЗУ	5
Секушин Е. В., Осипенко В. И., Копыл П. А. — Универсальный программируемый контроллер прерываний	7
Грязнов Д. О., Омеляненко А. Х., Шевчук Ю. В. — Электронный диск в вычислительном комплексе под управлением ОС, совместимой с ОС UNIX	12
Хохлов Д. А., Масленков М. Я., Крюков М. Г. — Использование БИС KP580ГФ24 в задающем генераторе дисплейного модуля	13
Алешин А. Н., Крюков С. Н. — Контроллер КНМЛ для микроЭВМ «Электроника 60»	14
Канцеров В. А., Родионов В. В., Стригин В. Б., Халько Н. Н. — Устройство сопряжения дисководов EC5074 с контроллером дисководов PLX45D	23
Гайдучок Р. М., Шумский Г. А. — Символьно-графический дисплей на базе газоразрядной индикаторной панели	26
Новиков А. В., Климов В. П., Смирнов В. Н., Ковальков В. К. — Устройство последовательного интерфейса мультиплексного канала	27
Котельников Ю. Н., Крочев А. М., Куликов А. С., Никитин С. П., Обходов Е. А. — Технологические контроллеры серии УТК6	33
Корнев В. А., Пройдаков Э. М., Щелкунов Н. Н., Дианов А. П. — Универсальный контроллер на базе ОЭВМ KP1816BE51	37
Вуколиков В. М., Дроздов Ю. Н., Колобков В. В., Лазарев А. В., Панкрац Е. П. — Пакет программ управления приборами через магистраль МЭК 625.1 для ПЭВМ	43
Мамджян Г. Г., Наумчик М. П., Прохоренко А. Н., Розман Б. Я., Горбов Д. С., Рузанова Н. И. — Комплекс микропроцессорных модульных средств	46
Белоусова Г. В., Болмазов И. В., Кулаков В. Б., Несветайлов В. В., Федорова Н. А. — Инструментальная система отладки программ	47
Дмитриев В. А., Бедарев А. В., Гравов В. С., Бурнашев А. Д., Сапо С. Н. — Стенд для изучения временных соотношений сигналов	53
Падиряков Ю. А. — Текстовый экраный редактор SED для дисплея «Электроника MC 7401»	55
Вельтмандер П. В., Власов С. И. — ОФОРИС — прикладная графическая система формирования иллюстративных материалов	58
Алексеев Г. И., Мыльников С. П. — Интерактивная графическая система ГРАФ	61
Маркова Н. А. — Графический стандарт и его внедрение на ПЭВМ	65
Данильченко Н. Т., Лозовой Л. Н. — Перенастраиваемая микропроцессорная система управления самоходным рентгеновским дефектоскопом «Сирена-1»	69
Матвеев В. Ф., Никитюк В. А., Прокопенко В. К., Шевцов П. И. — Программируемый контроллер на базе однокристалльной ЭВМ KM1816BE48 для группового управления станками с ЧПУ в ГПС	73
Анашкин И. И., Гагарин В. А. — Многофункциональный информационно-управляющий комплекс на базе одноплатных ЭВМ	79
Лосев И. С. — Будущее компьютерных игр	82
Липаев В. В. — Некоторые особенности подготовки специалистов по проектированию программных средств	85
Карась И. З. — Работы по созданию законодательства информатики в СССР	88
Интерфейсные БИС микропроцессорного комплекта K1801: микросхема K1801BP1-035	89

© Московский экспериментальный вычислительный центр «Алекс»

УДК 681.3.068

В. Г. Артюхов, В. А. Кондратюк, А. И. Макеенок,
В. И. Пенской, Е. В. Стацюра

ЦИФРОВОЙ ПРОЦЕССОР КМ1813ВЕ1: КОМПЛЕКС СРЕДСТВ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

Комплекс может использоваться как кросс-система проектирования программного обеспечения СБИС КМ1813ВЕ1 или как АРМ для проектирования и отладки систем цифровой обработки сигналов (ЦОС) на ее основе.

Общая программа для СБИС комплектуется из подпрограмм, реализующих отдельные блоки, путем их размещения по адресам РПЗУ в требуемой последовательности и организации взаимосвязи между отдельными подпрограммами назначением общих ячеек СОЗУ для входных и выходных сигналов блоков. Эта задача реализуется штатными редакторами текстов. Для трансляции используется программа ассемблера.

В процессе отладки программы на программной модели исследуется реакция проектируемого устройства на воздействия тестовых сигналов, задаваемых в табличной или аналитической форме. Такой подход позволяет проверять работу устройства на сигналах, которые трудно получить аппаратным путем, а также визуализировать промежуточные результаты работы процессора. На этапе комплексной отладки исследуются: работа в реальном масштабе времени, реакция на реальные входные сигналы и взаимодействие с внешней аппаратурой.

Программное обеспечение комплекса включает (рис. 1): библиотеку типовых программ RENVLIB; пакет прикладных программ проектирования цифровых фильтров RENVFIL; кроссассемблер RENASM; программы — моделирования устройств на основе

ЦПОС RENOTL, управления программатором RENPRG, обмена информацией с устройством РАСТР RASTR.

Библиотека RENVLIB насчитывает 47 программ, в том числе программы реализации типовых подсистем обработки сигналов — цифровых фильтров (ЦФ) для широко распространенных структур с постоянными и изменяющимися коэффициентами, анализатора спектра сигналов, автокоррелятора, элементов систем авторегулирования, систем обработки речи и др.

Пакет RENVFIL разработан на основе ППП САРАМ* с учетом возможностей ЦПОС по реализации ЦФ и включает следующие программы:

- проектирования рекурсивных ЦФ по аналоговому прототипу;
- оптимизационного проектирования рекурсивных ЦФ с произвольными АЧХ;
- проектирования не рекурсивных ЦФ с линейной фазовой характеристикой.

Каждая программа позволяет также генерировать программу реализации спроектированного ЦФ с возможностями выбора способа кодирования коэффициентов ЦФ в двоичном или каноническом знако-цифровом кодах (КЗЦК), повторной генерации программы после изменения разрядности коэффициентов, если длина программы больше допустимой.

Исходные данные вводятся и корректируются в диалоговом режиме с терминала. После расчета и вывода на печать коэффициентов передаточной функции ЦФ, ее нулей и полюсов осуществляется построение частотных характеристик в линейном и логарифмическом масштабах при заданной разрядности коэффициентов, а также временной анализ — построение импульсной и переходной характеристик. Временные характеристики вычисляются путем моделирования ЦФ с заданной разрядностью коэффициентов и результатов арифметических операций. Предусмотрена возможность многократного повторения моделирования при различных значениях разрядностей и параметров входных сигналов. Результаты выводятся на экран терминала и печатающееся устройство в виде графиков и таблиц. Если результаты удовлетворяют требованиям технического задания, осуществляется генерация программы, реализующей спроектированный ЦФ.

Кроссассемблер RENASM предназначен для трансляции и диагностики программ, написанных на языке ассемблера КМ1813ВЕ1, формирования файлов с листингом и объектным кодом транслируемых программ. Программа RENASM использует три псевдокоманды: TITLE, ORG, END и четыре типа форматов команд: ALU DST SRC [SHIFT] [IOI], RNZ [IOI], JNZ [IOI], IOI, где

- ALU — операции ALU;
- DST — имя ячейки ОЗУ, являющейся источником одного из операндов и приемником результата операции, или имя регистра DAR;
- SRC — имя ячейки ОЗУ — источника второго операнда, имя регистра DAR или имени констант КР0...КР7, КМ1...КМ8 (имена операций ЦПОС и псевдокоманд не могут быть использованы в качестве имен ячеек ОЗУ);

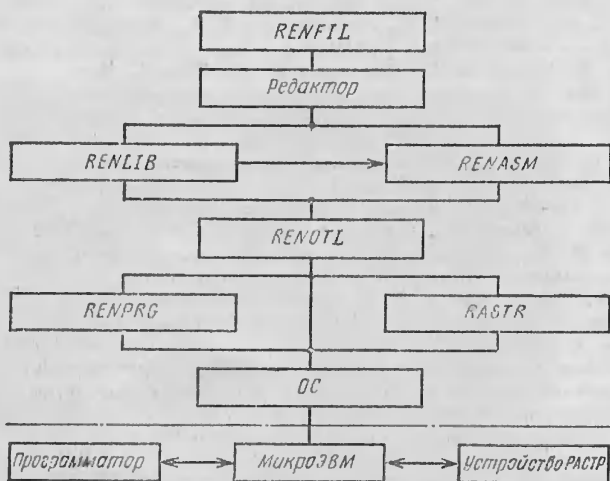


Рис. 1. Состав комплекса средств автоматизации проектирования систем на СБИС КМ1813ВЕ1

* Артюхов В. Г., Брытов А. А., Кондратюк В. А. Пакет прикладных программ автоматизированного проектирования цифровых фильтров на СМ ЭВМ // Автоматизация проектирования в электронике: Респ. межвед. науч.-техн. сб. — Киев: Техника. — 1985. — Вып. 32. — С. 73—79.

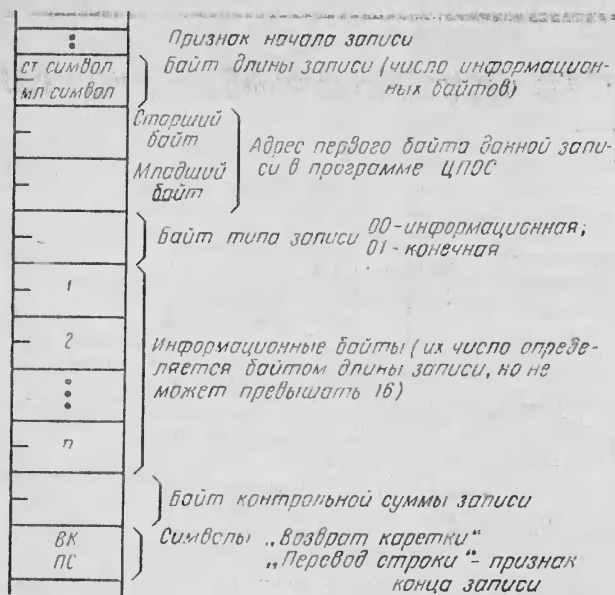


Рис. 2. Структура одной записи объектного файла:

каждый байт записывается в виде двух символов в коде КОИ7; контрольная сумма вычисляется как дополнение до двух суммы всех байтов записи, вычисленной по модулю 256, т. е. без учета переноса

SHIFT — операция сдвига R00...R13, L01, L02 (во всех случаях обязательно использовать двузначное число для обозначения числа сдвигов);

RNZ, JNZ — команды условного перехода соответственно в начало программы и на 32 ячейки вперед;

IOI — аналоговая операция.

Поля, заключенные в квадратные скобки, могут отсутствовать. В этом случае им присваиваются значения, соответствующие команде NOP. Если поле IOI является единственным в команде, то оно не может иметь значения CND5 и CND7-0.

Если после поля IOI присутствует текст, отличный от комментария, то транслятор его не обрабатывает, но выдает предупреждающее сообщение.

В процессе трансляции RENASM обнаруживает и указывает семь типов некорректных ситуаций в программе ЦПОС:

ошибки переполнения памяти команд, положения команд, расположения в недопустимых адресах РПЗУ, переполнения ОЗУ;

синтаксические ошибки в тексте программ; предупреждение о нестандартных сочетаниях полей команды и нестандартных последовательностях команд.

Ошибки и предупреждения указываются в первых четырех позициях строки листинга программы ЦПОС. Первый символ — тип ошибки или предупреждения, второй — всегда пробел, а последние два символа содержат информацию о поле команды, в котором обнаружена ошибка, или о смысле предупреждения.

Программа RENASM создает файл с объектным кодом программы ЦПОС в 16-ричном (HEX) формате, принятом в ЭВМ СМ1800, МСУВТ В7, системах отладки С0, К1-30, устройстве РАСТР.

Объектный файл представляет собой информационную последовательность (рис. 2), заканчивающуюся записью вида

:00 0000 01 FF <ВК> <ПС>

При создании объектного файла каждая команда программы ЦПОС, содержащая 24 разряда, представляется в виде трех байтов, записываемых последова-

тельно. Минимальной адресуемой единицей информации также является байт, а не команда ЦПОС. Объектный файл содержит коды только тех команд ЦПОС, которые определены в транслируемой программе. Неиспользуемые ячейки РПЗУ ЦПОС никаким кодом по умолчанию не заполняются.

Если в программе ЦПОС имеются ошибки, то объектный файл не создается. Предупреждения, выдаваемые программой RENASM, носят информационный характер и не влияют на выдачу объектного кода.

Программа моделирования RENOTL предназначена для моделирования систем, включающих до восьми БИС КМ1813ВЕ1 и до четырех источников входных сигналов.

Она позволяет:

назначать и оперативно изменять систему связей между БИС и источниками входных сигналов, т. е. структуру моделируемой системы;

задавать, анализировать и оперативно изменять описание четырех входных и трех выходных сигналов моделируемой системы, индивидуально сохранять их на диске под выбранными именами и считывать с диска;

устанавливать любой из моделируемых ЦПОС или источников входных сигналов или все одновременно в исходное состояние;

задавать и оперативно изменять совокупность условий в программах различных ЦПОС, после выполнения которых необходима индикация промежуточных результатов моделирования (точки индикации);

моделировать работу системы с выдачей результатов на дисплей или АЦПУ;

анализировать и, при необходимости, модифицировать содержимое встроенного ОЗУ любого из моделируемых ЦПОС;

моделировать работу системы с сохранением отчетов сигналов в трех точках системы, заданных первыми тремя точками индикации. Последовательности этих отчетов образуют три выходных сигнала моделируемой системы.

При описании структуры моделируемой системы используются модели входных сигналов, показанные на рис. 3, а. Параметры источника сигнала задаются при описании данного входного сигнала. Это дает возможность создать на диске библиотеку описаний типовых сигналов и использовать их при отладке различных систем на ЦПОС.

Входные сигналы моделируемой системы могут быть двух типов: 1 — отсчет-ввод, 2 — отсчет-время.

Сигналы типа 1 описываются только амплитудой без указания временных характеристик. При моделировании для каждой операции ввода выбирается один отсчет.

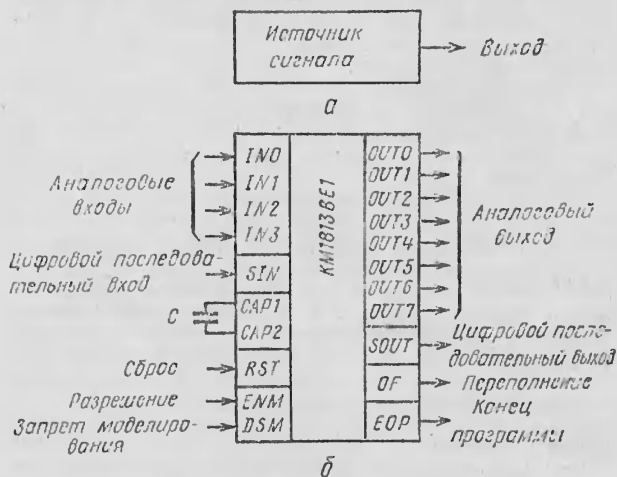


Рис. 3. Модель элементов в программе RENOTL

Несколько идущих последовательно команд IN рассматриваются как одна операция ввода. После выборки последнего отсчета выбирается первый отсчет, т. е. сигнал периодически повторяется.

Сигналы типа 2 описываются амплитудой отсчетов и временем появления каждого отсчета, значение сигнала между этими отсчетами (во времени) определяется при помощи линейной аппроксимации. Последний отсчет задает период повторения входного сигнала, а амплитуда должна быть равна амплитуде первого отсчета.

В моделируемой системе может быть определено до четырех входных и трех выходных сигналов. Выходные сигналы всегда бывают типа 2. Отсчеты всех сигналов хранятся в общем массиве, вмещающем 1024 отсчета. Описания входных сигналов могут быть введены или модифицированы оператором с клавиатуры терминала после включения этих сигналов в описание структуры моделируемой системы (задания их имен). Каждый процессор моделируется объектом, представленным на рис. 3, б. Входы IN0...IN3 и выходы OUT0...OUT7, OF непосредственно моделируют соответствующие выводы реальной БИС. Вход SIN и выход SOUT — процесс последовательного ввода-вывода информации. Хотя в реальной БИС эти выводы совмещены с входами IN0 и IN3, им даны отдельные имена для удобства пользования и простоты моделирования. Вход RST и выход EOP моделируют вывод RST/EOP реального процессора. В модели они также разделены.

Выводы WR и RD реального процессора, а также нагрузочные резисторы не описываются и не моделируются. Считается, что все внешнее, по отношению к процессору, оборудование моделируемой системы заведомо исправно и функционирует правильно.

Входы модели ENM и DSM, отсутствующие в реальном процессоре, предназначены для моделирования систем, в которых один процессор может управлять тактовыми сигналами, поступающими на другой, и, таким образом, приостанавливать выполнение его программы. Входы ENM и DSM являются как бы входами SR триггера. Поступление сигнала Лог.0 на вход DSM останавливает моделирование выполнения его команд, а сигнала Лог.0 на вход ENM разрешает продолжение моделирования. Вход DSM обладает более высоким приоритетом по отношению к ENM. К моменту начала моделирования считается, что моделирование выполнения команд во всех процессорах разрешено. В версии 1.0 программы RENOTL к входам ENM, DSM и RST можно подключать только выход EOP.

Каждый процессор моделируемой системы имеет уникальное имя, соответствующее имени файла с программой этого процессора на диске, и описывается следующими параметрами:

- программа процессора;
- режим ввода-вывода;
- время выполнения команды;
- величина накопительной емкости;
- переменными состояниями:
- содержимое ОЗУ;
- содержимое внутренних регистров;
- значение сигнала на внешних выводах;
- момент окончания выполнения текущей команды.

При передаче аналоговых сигналов используется модель, представленная на рис. 4, а. Опорное напряжение всегда считается равным 1 В. Коэффициент усиления усилителя задается в описании структуры моделируемой системы. При передаче аналоговых сигналов в модели учитывается, что по команде OUT на соответствующий выход процессора передается значение, составляющее 0,9 от значения, находящегося в регистре DAR.

При моделировании команд вывода аналоговых сигналов учитывается постоянная времени выходного устройства выборки и хранения (УВХ) с параметрами $R=1,5$ кОм и $C=500$ пФ. При моделировании команд

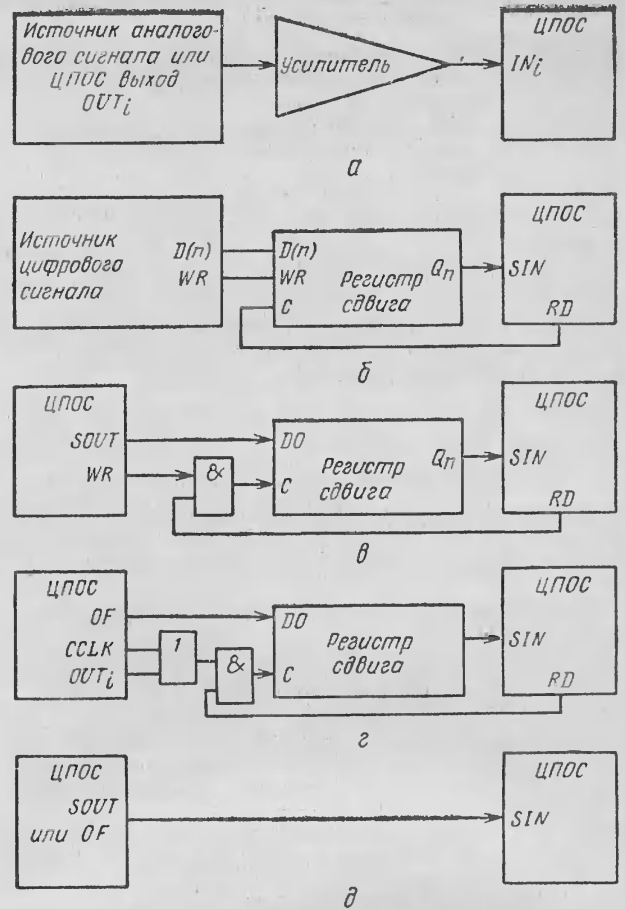


Рис. 4. Модели передачи сигналов в программе RENOTL

аналогового ввода, если источником сигнала является другой процессор или сигнал типа 2, учитывается постоянная времени входного УВХ с параметрами $R=1,5$ кОм, а C задается оператором. При моделировании вывода сигналов с TTL-уровнями на соответствующий выход процессора передаются значения 0,2 В, если в DAR находится величина меньше 0,5, и 4,0 В — в противном случае.

При последовательном вводе-выводе используются четыре различные модели.

1. Внешний источник сигнала загружает в сдвиговый регистр N -разрядное слово, а затем процессор вводит это слово последовательно по одному биту (рис. 4, б). Внешний источник сигнала, работающий в последовательном коде, моделируется путем использования одноразрядного слова.

2. Два процессора могут обмениваться информацией через сдвиговый регистр (рис. 4, в), моменты записи и чтения которого должны быть разнесены во времени.

3. Процессор-источник может выдавать информацию не только через выход SOUT, но и через выход OF (рис. 4, г). При этом для синхронизации сдвигового регистра используется сигнал CCLK, стробируемый сигналом OUT₁, который должен работать в режиме TTL и иметь значение Лог.0. Бит, записываемый в сдвиговый регистр, определяется наличием переполнения в команде с адресом вида $4N-1$. Такой способ последовательного вывода менее удобен, чем вывод через выход SOUT, однако он позволяет сохранить возможность ввода аналоговых сигналов, что существенно во

многих применениях. Разряд сдвигового регистра, с которого снимается сигнал на вход SIN, задается оператором при описании структуры системы. Максимальная разрядность сдвигового регистра 28 бит.

4. Если используемый разряд сдвигового регистра не указан, то подразумевается прямая передача с выхода SOUT или OF одного процессора на вход SIN другого процессора (рис. 4, д). При этом предполагается, что оба имеют одинаковое время выполнения команды и работают полностью синхронно (команда в команду).

Следует отметить, что значение сигнала на выходе OF определяется результатом выполнения арифметической операции в предыдущей команде.

Программа управления программатором RENPRG позволяет выполнять запись объектных кодов, отлаженных программ БИС КМ1813ВЕ1 во внутреннее РПЗУ, проверку правильности записи, чтение программ из РПЗУ. Работает под управлением ОС ДВК и имеет пять режимов: запись, чтение, сравнение программы, контроль импульсов записи, выход.

Программа RENPRG использует стандартный программатор РПЗУ типа МП-2, подключаемый к системной шине микроЭВМ. Для программирования РПЗУ СБИС КМ1813ВЕ1 необходимы кабель, соединяющий устройство управления и переходное устройство программатора, и заглушка с нагрузочными резисторами.

Программа обмена с устройством РАСТР предназначена для передачи объектных кодов отлаженных программ КМ1813ВЕ1 в ОЗУ устройства РАСТР, а также для приема программы из устройства РАСТР. Она проверяет правильность передачи и приема программ. Устройство РАСТР должно находиться в режиме «связь с ЭВМ». Программа имеет четыре режима работы: передача, чтение, сравнение программы в РАСТРе с программой на диске и выход.

Связь микроЭВМ с устройством РАСТР осуществляется через интерфейс ИРПР. Комплекс средств реализован на базе микроЭВМ «Электроника 60М» и ДВК. Программная часть комплекса в полном объеме поставлена на мини-ЭВМ СМ3, СМ4, СМ1420 в ОС РАФОС и ОС РВ. Программы RENFIL, RENASM, RENOTL написаны на языке Фортран-IV и могут быть поставлены на любой ЭВМ, имеющей соответствующий транслятор.

Организация базового рабочего места на ДВК2М показана на рис. 5. В составе АРМ используются штатные для ДВК устройства микроЭВМ «Электроника МС1201», дисплей «Электроника 15 ИЭ-00-013», накопитель на гибких магнитных дисках «Электроника ГМД-70 (ГМД 7012)» или «Электроника НГМД-6022», печатающее устройство DZM-180 (или любое другое штатное), а также дополнительные устройства — программатор МП2 или устройство РАСТР. Программатор используется для записи программ в РПЗУ процессора, а устройство РАСТР, кроме того, позволяет выполнять отладку на тестовых или реальных сигналах. Программатор МП2 подключается к системной шине ДВК2М, а устройство РАСТР — через интерфейс ИРПР.

Рассмотренные средства автоматизации проектирования позволяют выполнять полный комплекс работ по проектированию систем обработки сигналов на БИС КМ1813ВЕ1, легко реализуются на базе серийного оборудования. Открытая архитектура средств даст возможность изменять, наращивать аппаратные средства и дополнять состав программного обеспечения другими разработками, совместимыми по структуре выходного файла, адаптировать его под определенный класс задач: включать в состав АРМ пакеты проектирования алгоритмов цифровой обработки для конкретной прикладной области;

полнять библиотеку RENLIB типовыми программами конкретной прикладной области;
создавать средства формирования входных сигналов модели, характерных для данной области;
формировать средства интерпретации выходных сигналов модели устройства и представления результатов в форме, привычной разработчику.

252056, Киев-56, проспект Победы, 36 КПИ-2202,
кафедра систем автоматизации проектирования;
тел. 441-14-53

Статья поступила 31.03.88

УДК 681.327.8.06

А. А. Бондаренко, В. Ф. Скороходов

АСИНХРОННЫЙ КОНТРОЛЛЕР ДИНАМИЧЕСКОГО ОЗУ

Широко распространенный интерфейс И-41 допускает использование 8 и 16 разрядных задатчиков. Это, например, процессоры на основе МП КР580ВМ80 и КМ1810ВМ86. От основных модулей системы, в том числе и от ОЗУ, требуется возможность работы как с одним, так и с другим процессором. Контроллеры динамического ОЗУ [1, 2] не удовлетворяют данному требованию, так как являются полусинхронными (для их работы необходима косвенная информация о состоянии процессора: Ф2 в [1], CLK и ALE в [2]). Предлагаемый асинхронный контроллер динамического ОЗУ свободен от этих недостатков. Его работа синхронизируется сигналом CCLK частотой 10 МГц, который является асинхронным по отношению к работе процессора. Запрос на регенерацию фиксируется срезом сигнала CCLK, а обращение к ОЗУ — фронтом. При этом достигается асинхронность работы контроллера: запрос на регенерацию и обращение к ОЗУ никогда не могут появиться одновременно. Контроллер работает в режиме «прозрачной» регенерации и не требует никакой информации о состоянии процессора.

Однако такая универсальность контроллера приводит к определенным аппаратным затратам. Наряду с традиционными узлами контроллера (счетчиками регенерируемой строки, мультиплексорами адреса, узлом синхронизации) необходимо устройство, способное разрешать конфликты при одновременном запросе ОЗУ процессором и узлом регенерации — арбитр. Поэтому пред-

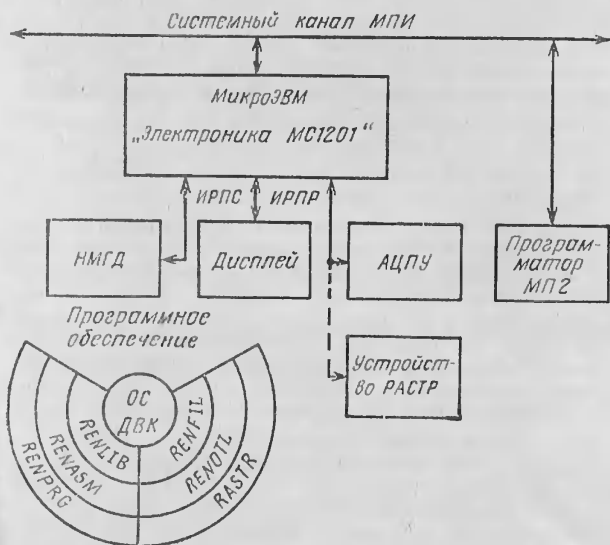


Рис. 5. Состав АРМ проектирования систем на СБИС КМ1813ВЕ1

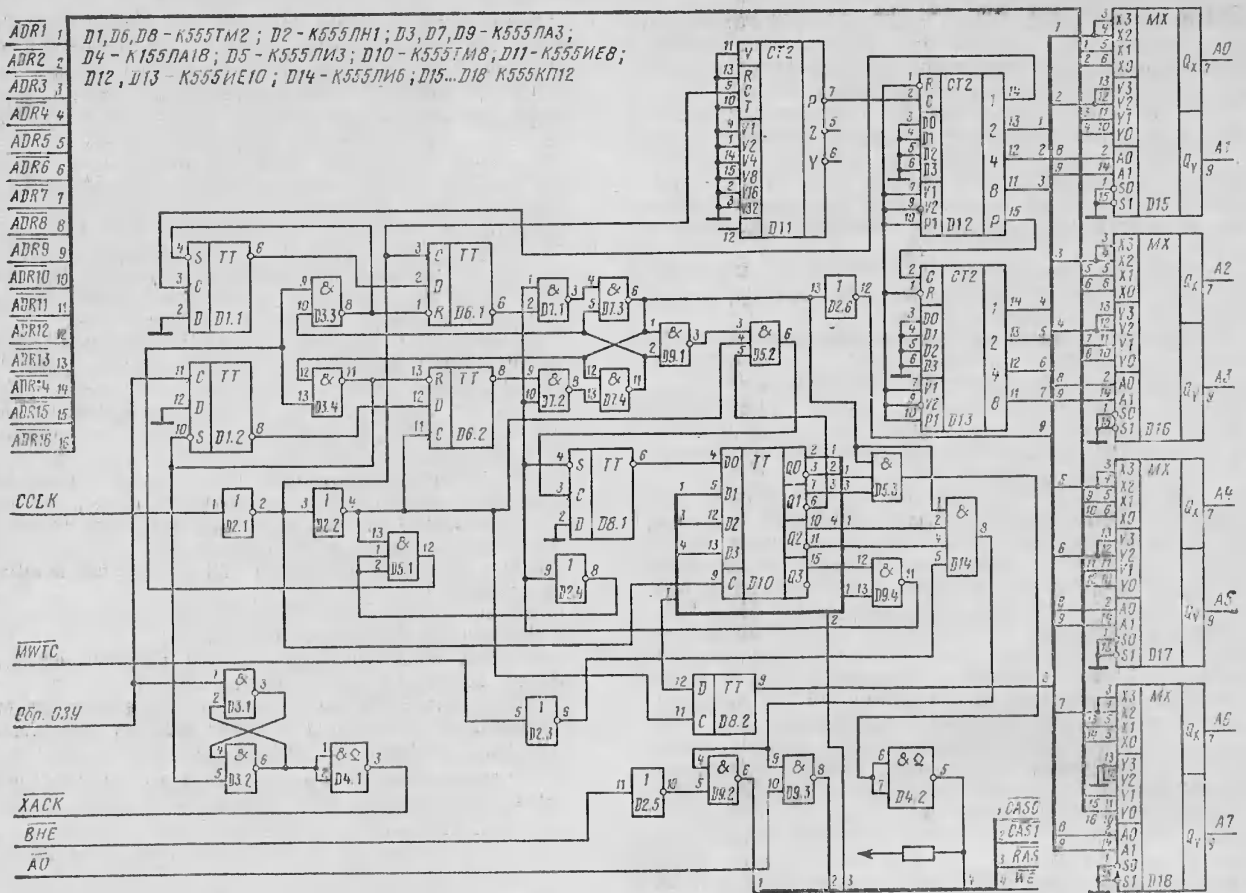


Рис. 1. Принципиальная схема контроллера

лагасмый контроллер динамического ОЗУ целесообразно использовать при построении микропроцессорных систем по модульному принципу. При проектировании вычислительных устройств, в которых будет использоваться один тип МП, например одноплатные микроЭВМ, выгоднее применять синхронные и полусинхронные контроллеры [1, 2].

Контроллер состоит из следующих узлов (рис. 1): формирователя адреса регенерируемой строки (D12, D13), мультиплексора, обеспечивающего подачу на микросхемы ОЗУ адреса регенерируемой строки и адреса с системной шины (D15, D16, D17, D18), формирователя сигналов $\overline{WE0}$ и $\overline{WE1}$ (D14, D2.3, D9.2, D9.3, D2.5), узла формирования сигнала готовности \overline{XACK} (D3.1, D3.2, D4.1), узла управления мультиплексором (D2.6, D8.2), узла арбитра (D1, D2.2, D3.3, D3.4, D6, D7, D5.1) и узла синхронизации (D5.2, D8.1, D10, D5.3, D4.2, D2.4, D9.4).

Регенерация элементов динамического ОЗУ происходит постоянно с периодом 16 мкс. При этом общее время регенерации для 128 строк не превышает допустимое [3]. Временные диаграммы цикла регенерации приведены на рис. 2, а. Тактирующие импульсы CCLK поступают на предварительный делитель частоты D11. Сигнал запроса на регенерацию формируется на младшем разряде D12. На остальных выходах этого счетчика, а также D13 формируются адрес регенерируемой в данный момент строки. Сигнал запроса на регенерацию запоминается в триггерах D1.1 и D6.1 (в триггере D6.1 по срезу тактирующих импульсов CCLK выход

$\overline{Q} = \text{Лог. 0}$). Если в данный момент отсутствует цикл обращения к памяти (на выходе элемента D7.4 Лог. 1), триггер режима D7 принимает состояние регенерации (на выходе элемента D7.3 Лог. 0). Сигнал с выхода элемента D7.3 через инвертор D2.6 поступает на вход A1 мультиплексоров D15...D18, устанавливая их для передачи адреса регенерируемой строки в элементы ОЗУ. Наличие низкого уровня на одном из выходов триггера режима D7 приводит к состоянию Лог. 1 на выходе \overline{Q} триггера D8.1 и запуску узла синхронизации. При этом на выходе $\overline{Q0}$ регистра D10 формируется сигнал \overline{RAS} . Через четыре такта сигнала CCLK Лог. 0 с выхода элемента D9.4 выход \overline{Q} триггера D8.1 будет равен Лог. 0, что приведет к запрещению работы узла синхронизации и возвращению его в начальное состояние. Следующий запуск узла синхронизации возможен только при наличии на выходе $\overline{Q1}$ регистра D10 Лог. 1, что произойдет не ранее, чем через два такта сигнала CCLK. Это позволяет обеспечить необходимую задержку между циклами регенерации и обращения к ОЗУ. Сигнал Лог. 0 с выхода элемента D9.4 поступает через элемент D2.4, D5.1, D3.3 на вход \overline{R} триггера D6.1 и на вход \overline{S} триггера D1.1, что приводит к сбросу обслуженного запроса на регенерацию.

При обращении к ОЗУ сигнал высокого уровня с дешифратора адреса разблокирует схему формирования готовности \overline{XACK} и запоминается в триггерах D12 и D6.2 (в триггере D6.2 по фронту тактирующих импульсов CCLK). Если в данный момент отсутствует цикл регенерации, то триггер режима D7 переводится в режим

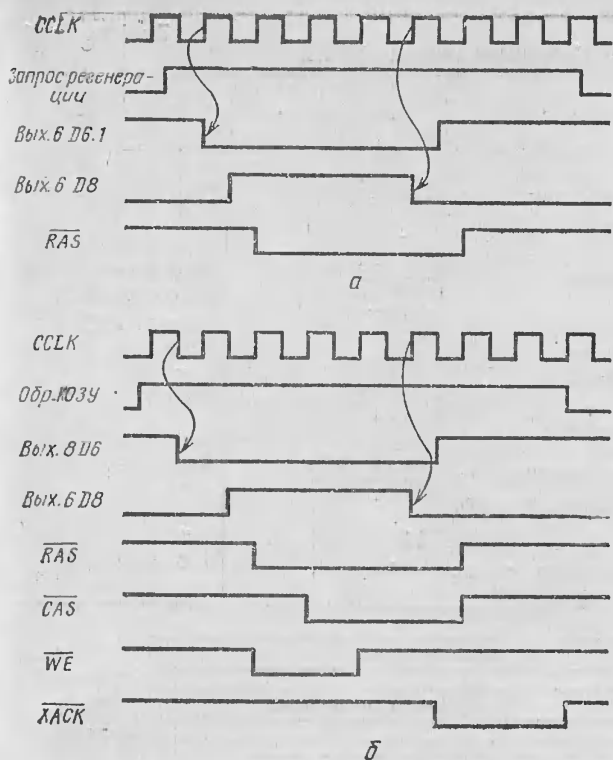


Рис. 2. Временные диаграммы:
а — режим регенерации; б — режим чтения

обращения к ОЗУ (на выходе элемента D7.4 Лог.0). Сигнал Лог.1 с выхода элемента D7.3 через инвертор D2.6 поступает на вход A1 мультиплексоров D15...D18; на вход A0 — Лог.0 с выхода триггера D8.2 (мультиплексоры передают в элементы ОЗУ младший байт адреса). Сигнал Лог.0 с выхода элемента D7.4 через элементы D9.1 и D5.2 устанавливает выход \bar{Q} триггера D8.2 в состояние Лог.1, что приводит к запуску узла синхронизации. По срезу первого такта CCLK (рис. 2, б) появляется сигнал \bar{RAS} . При записи цикла в ОЗУ (активный сигнал \overline{MWTC}) на выходе элемента D14 вырабатывается сигнал \overline{WE} , который поступает на формирователь сигналов $\overline{WE0}$ и $\overline{WE1}$. В зависимости от состояния сигналов $\overline{ADR0}$ и \overline{BHE} (запись: в старший —

нечетный байт, в младший — четный байт или в оба байта сразу) с выхода формирователя на микросхемы ОЗУ поступают сигналы $\overline{WE0}$ (запись в младший байт) и $\overline{WE1}$ (запись в старший байт). По фронту первого такта CCLK триггер D8.2 устанавливается в единичное состояние, мультиплексоры переключаются и передают в элементы ОЗУ старший байт адреса. По срезу второго такта сигнала CCLK на выходе D5.3 создается сигнал \overline{CAS} , который через D4.2 подается на микросхемы ОЗУ; по срезу третьего такта сигнал \overline{WE} сбрасывается; по срезу четвертого такта на выходе D9.4 устанавливается Лог.0, который переключает триггер D8.1 (при следующем такте CCLK сигналы \bar{RAS} и \overline{CAS} сбрасываются и узел синхронизации переходит в исходное состояние). Лог.0 с выхода D9.4 через D2.4, D5.1 и D3.4 по фронту четвертого такта CCLK переводит схему формирования сигнала готовности \overline{XACK} в активное состояние и сбрасывает триггеры D1.2 и D6.2.

Для правильного считывания данных их необходимо зафиксировать. Для этого наиболее удобно использовать регистры-защелки КР580ИР82 или КР580ИР83, подавая на вход STB сигнал \overline{CAS} .

Контроллер допускает построение модулей динамического ОЗУ большого объема, состоящих из нескольких линеек БИС ОЗУ.

Линейка БИС ОЗУ выбирается с помощью сигналов $\overline{CAS0}$... $\overline{CAS7}$, формируемых на выходах дешифратора (например, К555ИД7), работа которого разрешается сигналом \overline{CAS} (на его входах старшие разряды адреса).

Однако при этом следует обратить внимание на нагрузочную способность мультиплексоров и формирователя сигналов $\overline{WE0}$ и $\overline{WE1}$ и в случае необходимости предусмотреть развязку по выходам.

454044, Челябинск, пр. Ленина, 76, ЧПИ, кафедра радиотехнических систем; тел. 39-92-87

ЛИТЕРАТУРА

1. Еремин Ю. А., Морозов А. Г. Контроллер динамического ОЗУ для микропроцессорных устройств // Микропроцессорные средства и системы. — 1986. — № 3. — С. 75—77.
2. Иванов С. Н., Романов А. Ф., Чернышев Ю. Н. Одноплатная микроЭВМ на МПК БИС серии К1810 // Микропроцессорные средства и системы. — 1986. — № 3. — С. 8—13.
3. Микросхема динамического ОЗУ К58ЕРУ5. Справочная информация // Микропроцессорные средства и системы. — 1985. — № 3. — С. 91—94.

Статья поступила 26.02.88

УДК 681.326.35.77

Е. В. Секушин, В. И. Осипенко, П. А. Копыл

УНИВЕРСАЛЬНЫЙ ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР ПРЕРЫВАНИЙ

В микропроцессорных системах для обмена данными с периферийными устройствами применяется программно-управляемая передача данных. При этом операции ввода-вывода осуществляются с прерыванием выполняемой программы. Этот способ реализуется БИС КР1818ВН19, изготавливаемой нашей промышленностью,

Микросхема обслуживает до восьми индивидуально маскируемых запросов на прерывание, решает приоритеты и выдает полностью программируемые ответы до 4 байт на каждое прерывание. Конструктивно универсальный программируемый контроллер прерываний (МПКП) выполнен в пластмассовом корпусе

2121.28-5 (рис. 1, табл. 1).

Байты ответной реакции — полностью программируемые, поэтому может быть использована любая инструкция или протокол векторизации (работа с общим или отдельным вектором), соответствующие главному процессу.

Режимы работы микросхемы устанавливаются программно: с общим или отдельными векторами; разрешения или блокировки всех прерываний; прерывания или опроса; автоматического сброса подтвержденных прерываний (по выбору).

В микросхеме предусмотрены управление полярностью сигналов ввода-вывода прерываний и возможность

Назначение выводов УПКП

Номер вывода	Назначение вывода	Обозначение	Тип вывода
1	Выбор микросхемы	\overline{CS}	Вход
2	Запись	\overline{W}	"
3	Чтение	\overline{R}	"
4...11	Канал данных	D0...D7	Вход-выход с тремя состояниями
12	Ответ	ASW	Вход-выход
13	Каскадирование	CAS1	Вход
14	Общий вывод	GND	—
15	Пауза	PAUSE	Выход
16	Каскадирование	CAS0	"
17	Прерывание	INT	"
18...25	Запрос прерывания	IRQ0...IRQ7	Вход
26	Подтверждение прерывания	INTA	"
27	Адрес	A0	"
28	Вывод питания от источника напряжения	U _{cc}	5 В

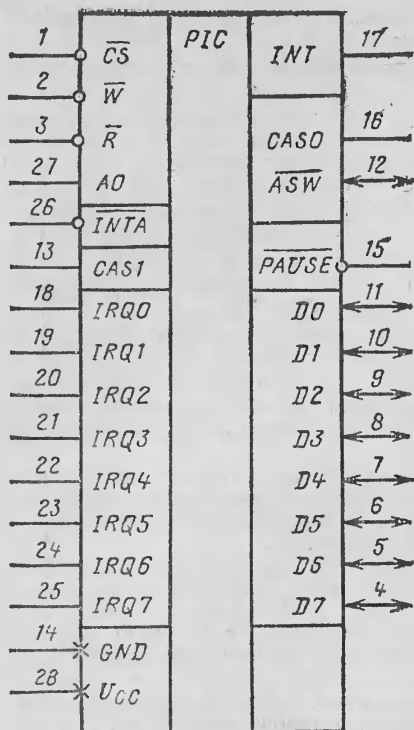


Рис. 1. Условное графическое изображение УПКП

запроса на прерывание средствами ПО.

В УПКП простая структура (рис. 2) неограниченного расширения числа обслуживаемых запросов с помощью каскадного соединения микросхем (последовательной цепи, определяющей приоритет устройств).

Обмен информацией с ЭВМ осуществляется по 8-разрядной двунаправленной шине данных.

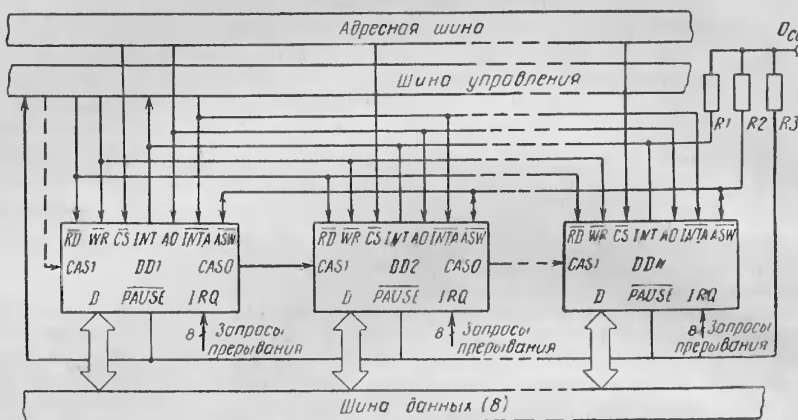


Рис. 2. Схема каскадного соединения УПКП

Основные электрические параметры КР1818ВН19 в диапазоне температур -10...70° С

Напряжение источника питания U _{cc} , В	5 ± 5%
Ток потребления I _{cc} , мА, не более	125
Входное напряжение, В:	
высокого уровня, не менее	2,0
низкого уровня, не более	0,8
Выходное напряжение, В:	
высокого уровня, не менее	2,4
низкого уровня, не более	0,4
Выходной ток, мА:	
высокого уровня, не более	-0,21
низкого уровня, не более	3,2
Емкость нагрузки по выходам, пФ, не более	100

Структура УПКП КР1818ВН19 организована на базе внутренней шины. В ее состав входят (рис. 3):

РЗП — регистр запросов прерываний. Его разряды устанавливаются при поступлении на соответствующий вход IRQ сигналов от внешних устройств, ожидающих обслуживания, или программно с центрального процессорного устройства (ЦПУ). Раз-

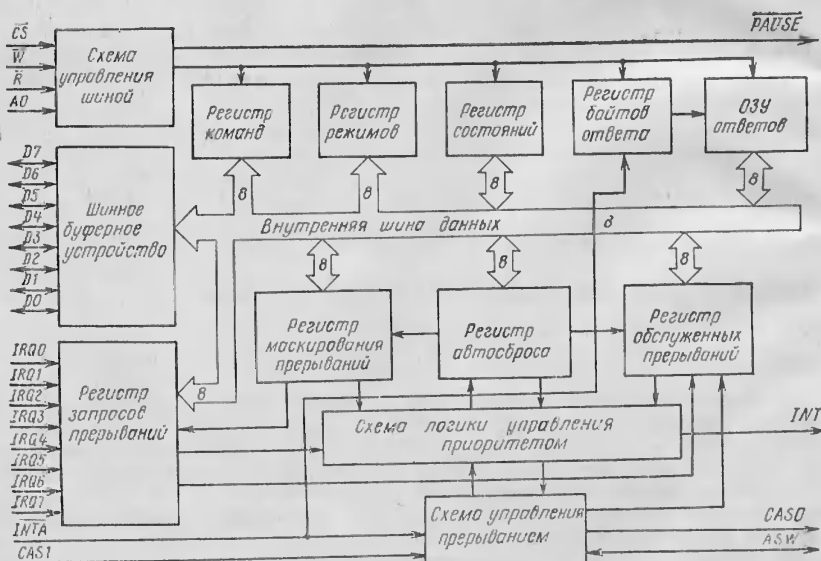


Рис. 3. Структурная схема УПКП

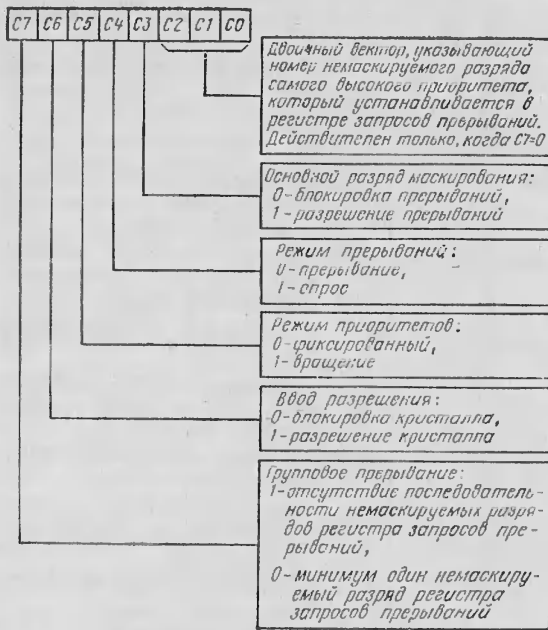


Рис. 4. Назначение разрядов регистра состояний

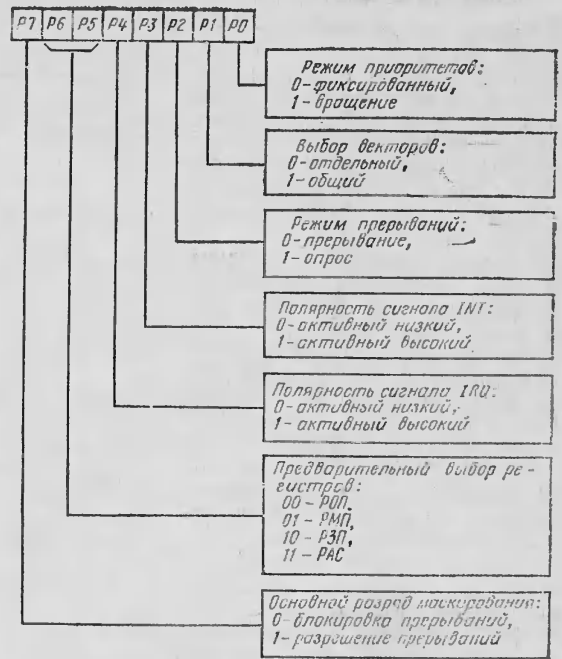


Рис. 5. Назначение разрядов регистра режимов

ряды РЗП могут сбрасываться программно и автоматически.

РОП — регистр обслуженных прерываний для хранения информации об обслуженных запросах, а также для блокирования всех запросов более низкого приоритета. Его разряды устанавливаются с помощью логики приоритета; соответствующий разряд РЗП при этом сбрасывается. Если запрос не запрограммирован на автоматический сброс, то соответствующий разряд РОП сбрасывается программно с ЦПУ для обслуживания запросов с более низким приоритетом. При установке исходного состояния сбрасываются все разряды РОП и РЗП;

РМП — регистр маскирования прерываний для разрешения или блокировки сигналов прерываний. Его разряды соответствуют входам IRQ. Регистры могут параллельно загружаться, устанавливаться и сбрасываться с помощью программного управления. В исходном положении все разряды РМП устанавливаются в состояние Лог. 1, блокируя прохождение всех запросов. Маскирование запросов не запрещает их запись в РЗП (такой запрос вызывает прерывание после сброса соответствующего разряда РМП). Инициализировать выход прерывания INT могут только немаскированные сигналы запросов.

РАС — регистр автосброса, содержащий один разряд для каждого входа IRQ. Он определяет рабочий режим для каждого из разрядов РОП. Если разряд РАС отключен, при подтверждении прерывания устанавливается соответствующий разряд РОП

(сбрасывается программно). Если разряд РАС установлен, соответствующий разряд РОП сбрасывается ап-

паратно после окончания последовательности подтверждения прерывания. Число байтов шлота, связанно-

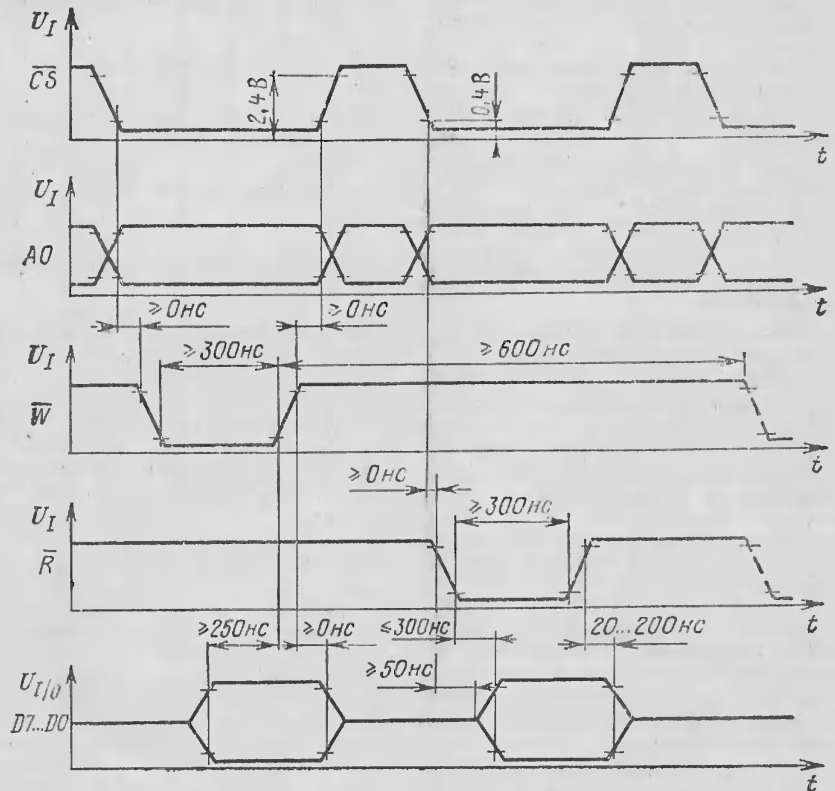


Рис. 6. Временные диаграммы работы КР1818ВН19 в режиме программирования и обслуживания по опросу

Команды загрузки

№ пп	Код команды									Команда
	D7	D6	D5	D4	D3	D2	D1	D0		
1	0	0	0	0	0	0	0	0	0	Установить исходное состояние
2	0	0	0	1	0	X	X	X		Сбросить все разряды регистров РЗП и РМП
3	0	0	0	1	1	B2	B1	B0		Сбросить разряд регистров РЗП и РМП, определенный с помощью B2, B1, B0
4	0	0	1	0	0	X	X	X		Сбросить все разряды регистра РМП
5	0	0	1	0	1	B2	B1	B0		Сбросить разряд регистра РМП, определенный с помощью B2, B1, B0
6	0	0	1	1	0	X	X	X		Установить все разряды регистра РМП
7	0	0	1	1	1	B2	B1	B0		Установить разряд регистра РМП, определенный с помощью B2, B1, B0
8	0	1	0	0	0	X	X	X		Сбросить все разряды регистра РЗП
9	0	1	0	0	1	B2	B1	B0		Сбросить разряд регистра РЗП, определенный с помощью B2, B1, B0
10	0	1	0	1	0	X	X	X		Установить все разряды регистра РЗП
11	0	1	0	1	1	B2	B1	B0		Установить разряд регистра РЗП, определенный с помощью B2, B1, B0
12	0	1	1	0	X	X	X	X		Сбросить разряд регистра РОП самого высокого приоритета
13	0	1	1	1	0	X	X	X		Сбросить все разряды регистра РОП
14	0	1	1	1	1	B2	B1	B0		Сбросить разряд регистра РОП, определенный с помощью B2, B1, B0
15	1	0	0	P4	P3	P2	P1	P0		Загрузить разряды 0..4 регистра режимов по определенной последовательности
16	1	0	1	0	P6	P5	0	0		Загрузить разряды 5, 6 регистра режимов по определенной последовательности
17	1	0	1	0	P6	P5	0	1		Загрузить разряды 5, 6 регистра режимов и установить разряд 7
18	1	0	1	0	P6	P5	1	0		Загрузить разряды 5, 6 регистра режимов и сбросить разряд 7
19	1	0	1	1	X	X	X	X		Предварительно выбрать регистр РМП для последующей загрузки с шины данных
20	1	1	0	0	X	X	X	X		Предварительно выбрать регистр автосброса для последующей загрузки с шины данных
21	1	1	1	BY1	BY0	L2	L1	L0		Загрузить BY1, BY0 в регистр байтов ответа и предварительно выбрать уровень памяти ответов, определенный с помощью L2, L1, L0 для последующей загрузки с шины данных

BY1, BY0 — разряды регистра байтов ответа, с помощью которого устанавливается число байтов ответа (00—1; 01—2; 10—3; 11—4 байта)

го с каждым прерыванием, программируется независимо, поэтому прерывания могут иметь ответы различной длины (хранятся в восьми 2-разрядных регистрах байтов ответа). Для каждого запроса УПКП принимает один, два, три или четыре INTA (по числу байтов в ответе) и формирует сигнал ASW низкого уровня.

ОЗУ ответов (8×32) для хранения до 4 байт информации для каждого из восьми входов IRQ. Все разряды памяти программируемые. Они обеспечивают ввод требуемого вектора, кода операции, инструкции и других данных. По сигналу низкого уровня на входе INTA УПК передает информацию о прерываниях из ОЗУ на шину данных.

РС — регистр состояний, особенно эффективен при работе в режиме опроса для определения устройств, вызывающих прерывание (рис. 4).

РР — регистр режимов, определяет режим работы микросхемы: РЗП, РОП, РАС или РМП для последующего считывания на шину данных (рис. 5). Пять разрядов РР (от 0 до 4) загружаются параллельно по команде, а 5, 6 и 7 — отдельными командами. РР не может считываться непосредственно на шину данных, но разряды 0, 2 и 7 (5, 4, 3) считываются как часть РС.

РК — регистр команд, запоминает последнюю введенную команду. В зависимости от кода операции он может выполнять внутренние действия или переходить в состояние ожидания для последующей загрузки с шины данных.

Схема логики управления приоритетом анализирует состояние РЗП, РОП, РМП и РАС, устанавливает режим приоритетов и уровень приоритета немаскированных запросов прерываний, определяет порядок их обслуживания.

Схема управления шиной определяет направление передачи информации между УПКП и ЦПУ по шине данных, а также ее обмен между регистрами.

ШБУ — двунаправленное шинное буферное устройство принимает информацию с шины данных и передает ее с помощью D0...D7. В режиме программирования в УПКП через ШБУ записываются команды, загружаются регистры и память ответов. В режиме векторного прерывания (после его подтверждения) по запросу из микросхемы на шину данных через ШБУ передаются байты ответной информации. В остальное время на выходе ШБУ установлено высокое сопротивление.

СУП — схема управления прерыванием; анализируя состояние сигналов ASW, CAS1 и сигналов, формируемых логикой управления приоритетом, разрешает или блокирует по-

ступление сигнала группового прерывания.

После установки сигнала INT и поступления INTA СУП позволяет заканчивать ответы на подтвержденное прерывание, блокируя запросы более высокого приоритета, которые появились в момент ответа на ранее пришедшее прерывание. При обслуживании запроса СУП генерирует на выходе CASO сигнал низкого уровня, используемый для блокировки систем прерывания других микросхем КР1818ВН19.

Работа БИС КР1818ВН19

Для выполнения требуемых операций контроллер прерывания устанавливается в исходное состояние по команде с ЦПУ или автоматически во время включения питания (влияния на память ответов и регистр байтов ответа не оказывается).

Регистр режимов определяет комбинации вариантов работы УПКП, которые могут осуществляться ЦПУ. Его содержимое сбрасывается по команде установки исходного состояния.

Состояние нулевого разряда регистра соответствует режиму приоритетов. В режиме фиксированного приоритета приоритет входам запросов присваивается на основании их физического местоположения в интерфейсе кристалла, при этом вход IRQ0 обладает самым высоким, а IRQ7 — самым низким приоритетом. В режиме вращения относительный уровень приоритета такой же, как и для фиксированного режима. Местоположение самого высокого уровня приоритета определяется относительно последнего обслуженного запроса.

Состояние первого разряда определяет вариант работы с отдельным или общим вектором. Отдельная векторизация обеспечивает определенное местоположение в ЗУ ответов для каждого запроса на прерывание. Вариант работы в общем векторном режиме всегда формирует ответный сигнал, соответствующий входу IRQ0, независимо от подтверждения запроса.

Состояние пятого и шестого разрядов указывает регистр, содержимое которого будет считываться при последующей операции чтения ($A0=0$, $\bar{R}=0$). Код выбранного регистра сохраняется до тех пор, пока не будет записан другой код или команда установки в исходное состояние.

Седьмой разряд — основной в маскировании. Он используется для блокировки всех входов запросов без изменения содержимого регистра РМП. Когда основной разряд маскирования сброшен, на выходе CASO формируется сигнал низкого уровня, поэтому в многокристалльных системах прерывания один основной разряд маскирования может блокировать всю структуру прерываний. Его состояние считывается в третьем разряде регистра состояния, который выбирается непосредственно через адресный

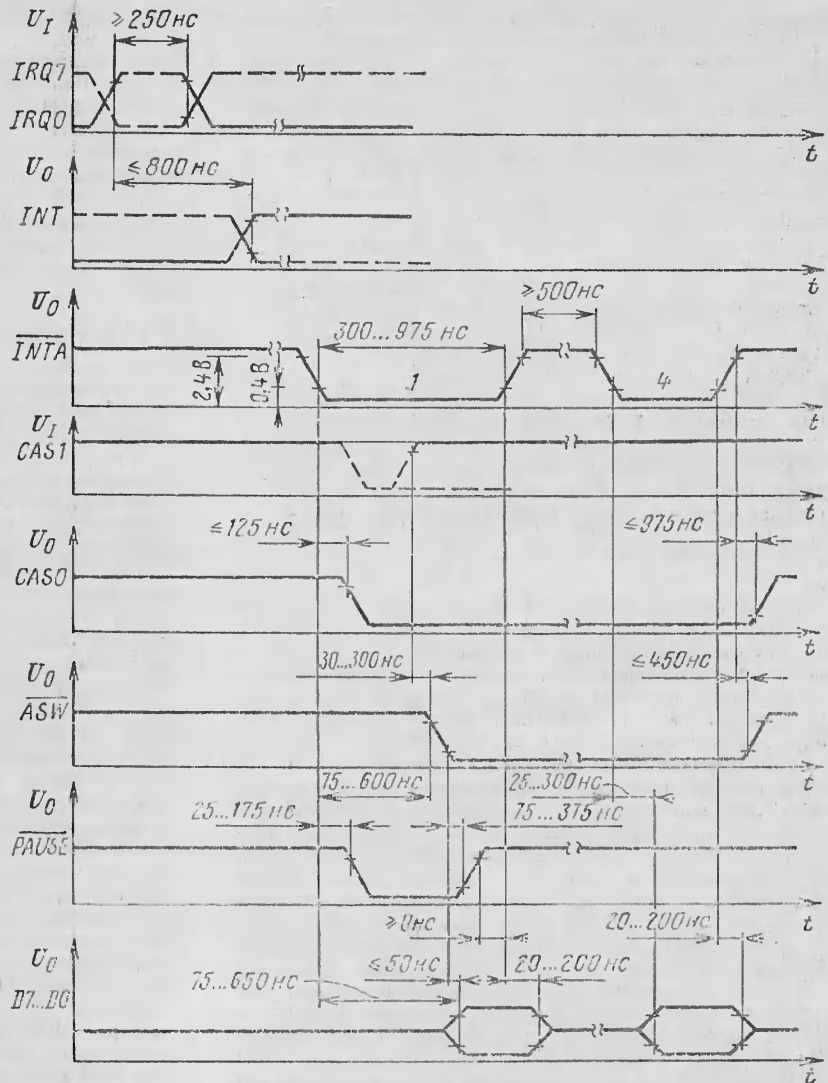


Рис. 7. Временные диаграммы работы КР1818ВН19 в режиме обслуживания по запросу

вход А0. Другие внутренние регистры считываются в два этапа: выбор соответствующего регистра при помощи пятого и шестого разрядов регистра режимов и непосредственное считывание содержимого регистра на шину данных. Содержимое ОЗУ ответов может считываться только импульсами INTA.

РМП и РАС загружаются в два этапа: первоначально подается команда выбора соответствующего регистра, затем осуществляется загрузка регистра с шины данных. Для загрузки каждого уровня памяти ответов выдается команда предварительного выбора соответствующего уровня (0...7) и его длины (1...4 байт). Затем для загрузки этого уровня выполняется соответствующее число операций записи данных (1...4 байт данных).

Выход PAUSE может быть использован ЦПУ для удлинения ци-

кла подтверждения и обеспечения автоматической регулировки управляющей синхронизации. Требуемая ширина импульса INTA зависит от нескольких переменных: рабочей температуры, внутренней логической задержки, числа УПКП и уровня приоритета подтверждаемых прерываний.

Длительность первого импульса INTA определяет переменные для задержки выбора запроса. PAUSE будет оставаться на низком уровне до выбора первого байта ответа — появления сигнала ASW низкого уровня. Обычно процесс внутреннего прерывания довольно скоростной, особенно для систем с одним КР1818ВН19, следовательно, PAUSE будет оставаться на низком уровне только в течение короткого времени, которое не вызывает увеличения временных интервалов INTA.

Если во время обслуживания текущего запроса появляется запрос более высокого приоритета, сигнал INT будет выдаваться УПК, но распознаваться и подтверждаться только в том случае, если ЦПУ отпирает вход прерываний. В случае подтверждения, будет устанавливаться соответствующий разряд РОП более высокого приоритета, а запросы вкля-

дываться один в другой гнездовым способом.

Первый этап работы — модификация РР и РА для установки конфигурации, требуемой для определенного случая использования УПКП. Затем загружается ОЗУ ответов и счетчик байтов применяемых уровней запросов. После этого контроллер прерываний может выполнять операции.

Управление УПКП, конфигурация его работы и проверка внутреннего состояния осуществляются с помощью команд непосредственной загрузки ПК (табл. 2).

Временные диаграммы работы микросхемы приведены на рис. 6, 7.

Телефон 442-95-18, Москва

Статья поступила 29.07.87

ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ

УДК 681.32

Д. О. Грязнов, А. Х. Омеляненко, Ю. В. Шевчук

ЭЛЕКТРОННЫЙ ДИСК В ВЫЧИСЛИТЕЛЬНОМ КОМПЛЕКСЕ ПОД УПРАВЛЕНИЕМ ОС, СОВМЕСТИМОЙ С ОС UNIX

Вычислительные комплексы на базе микроЭВМ «Электроника 60М», «Электроника НМС 01100.1» приобрели широкое распространение. К сожалению, их возможности не удается использовать в полном объеме, так как они не имеют быстрых устройств внешней памяти. Накопители на гибких магнитных дисках («Электроника ГМД 7012», «Электроника НГМД 6022»), входящие в комплект поставки, достаточно хорошо служат для долговременного хранения информации, но плохо подходят для использования в качестве системного устройства, т. е. для размещения часто используемых утилит, создания временных файлов редакторов текстов, компиляторов и тем более для свопинга. Для системного устройства необходима высокая скорость доступа к хранимой информации; после окончания сеанса работы ее сохранять не обязательно. Этим требованиям идеально удовлетворяет электронный диск (ЭД). По быстродействию такое устройство на два порядка превосходит гибкий диск и даже по сравнению с жестким диском может обеспечить существенное преимущество. На последнем сравнении стоит остановиться подробнее.

Время доступа к информации на диске (без учета затрат на считывание заголовка и контрольной суммы) определяется следующим образом:

$$T_D = TP + TO + TC \times N,$$

где TP — время позиционирования головок диска на заданную дорожку; TO — время, проходящее с момента установки головок на требуемую дорожку до момента, когда под головкой окажется требуемый сектор; TC — среднее время передачи одного байта информации, определяющееся скоростью вращения диска; N — число байтов, которые требуется передать в течение операции.

Для ЭД это выражение выглядит так:

$$T_D = TC \times N,$$

где TC определяется, главным образом, пропускной способностью канала ЭВМ, и поэтому для ЭД его не удается сократить более чем в 1,5–3 раза по сравнению с жестким. Если допустить, что TC жесткого диска и электронного равны, то преимущество электронного диска заключается в отсутствии TP и TO. Таким образом, оно наиболее ярко проявляется при

$$(TC \times N) \ll (TP + TO).$$

Это соотношение выполняется, когда по характеру работы требуются обращения к разным участкам диска для обмена сравнительно короткими порциями информации. Именно такой характер носит работа с диском, если на нем размещается файловая система с произвольным распределением блоков информации (например, файловая система ОС UNIX). Для иллюстрации

можно сказать, что у диска CM5400 TP=35 мс, TO=10 мс, TC=4 мкс и, следовательно, при обмене блоками по 512 байт TP+TO=45 мс, а TC×N=2 мс.

Наиболее простой способ организации ЭД — использование в качестве него части оперативной памяти ЭВМ. Отображение дисковой структуры на эту часть памяти осуществляется драйвером электронного диска. Однако такое решение применимо только в том случае, если ПО не использует всей оперативной памяти, что маловероятно, особенно при работе ЭВМ в мультипрограммном режиме. На рассматриваемых микроЭВМ размер ОЗУ настолько ограничен, что данный подход не может быть применен непосредственно, без увеличения размера ОЗУ. Для его наращивания на машинах с 16-разрядным адресным словом необходимо введение хотя бы несложного диспетчера памяти [3], а это связано с модификацией существующих узлов ЭВМ, что нежелательно.

Более приемлемым представляется реализация ЭД в виде периферийного устройства. Большинство таких реализаций имеют в строение ввода-вывода ЭВМ буферный регистр данных и адресный регистр, содержимое которого определяет, к какому участку электронного диска осуществляется доступ [3]. После каждого обращения к регистру данных содержимое регистра адреса автоматически увеличивается на единицу. Интересным расширением такого интерфейса является обмен по принципу «окна», расположенного в строение ввода-вывода ЭВМ. Через это окно можно осуществлять произвольный доступ к участку электронного диска, начальный адрес которого определяется регистром адреса, а размер равен размеру окна [1]. Этот вариант имеет преимущество при многократных обращениях к ЭД, локализованных в пределах размера окна. К сожалению, его трудно использовать на практике, так как большинство ОС ориентировано на обмен с диском блоками размером 512 байт.

Общий недостаток описанных интерфейсов — непосредственное участие процессора в операции передачи данных на всем ее протяжении. Особенно сильно он заметен в многопользовательских системах и в системах реального времени. Параллельность выполнения программы и обмена между оперативной памятью и ЭД может быть достигнута за счет реализации обмена в режиме прямого доступа к памяти. В этом случае от процессора требуется только загрузка в регистры контроллера адреса нужного блока на диске, числа байтов в обмене, адреса буфера оперативной памяти и направления передачи информации, после чего он может продолжать выполнение программы, в то время как контроллер будет заниматься передачей информации. После завершения операции контроллер генерирует прерывание.

ЭД состоит из контроллера и накопителя. Связь между ними осуществляется по параллельной магистрали. За счет такого построения обеспечивается повышенная мобильность устройства. Для переноса ЭД на другую машину требуется переработка контроллера, накопитель же никаким изменениям не подвергается. При измене-

ниях в конструкции накопителя не требуется модификация контроллера.

Контроллер — двухместная плата в конструктиве микроЭВМ «Электроника 60», содержащая 65 корпусов ТТЛ-микросхем малой и средней степени интеграции. В его функции входит обмен с оперативной памятью в режиме прямого доступа, формирование запросов к накопителю, а также генерирование прерываний по завершении операции либо при появлении ошибки.

В качестве накопителя используются промышленные блоки памяти «Электроника 256К» объемом 256 Кслов (16-разрядных), работающие с групповым корректирующим кодом. Число блоков в накопителе 1...16, таким образом, объем ЭД может варьироваться от 0,5 до 8 Мбайт.

Магистраль связи с накопителем состоит из 14 линий дискового адреса (адрес задается с дискретностью 512 байтов), 16 линий данных и 7 линий управления. Время доступа к блоку информации (512 Кбайтов) на электронном диске составляет 1,5—2 мс.

Рассмотренный ЭД объемом 1,5 Мбайт использовался в составе вычислительного комплекса на базе микроЭВМ «Электроника 60М», в который входил также накопитель на гибких магнитных дисках «Электроника ГМД 7012». Комплекс работал под управлением ОС МИКРОС 1.2, разработанной в лаборатории операционных систем ИПС АН СССР.

МИКРОС 1.2 — инструментальная, мультипрограммная многопользовательская мобильная ОС, предоставляющая пользователю большой и удобный набор средств разработки, отладки, сопровождения программ и подготовки документации. Она позволяет легко и удобно проверять корректность программ, отлаживать их в терминах исходного языка, обрабатывать файлы различного содержания, редактировать и выполнять форматирование текстов, подготовку статей, книг, построение информационно-справочных систем. Разработчику ПО ОС МИКРОС 1.2 предоставляет широкий диапазон возможностей — от экранного редактора текста и компилятора с языка программирования Си до системы поддержки разработки и сопровождения больших программных комплексов и компилятора компиляторов. Интерпретатор командного языка ОС МИКРОС 1.2 имеет все возможности языка программирования высокого уровня, удобные средства создания конвейеров, программных каналов. ОС МИКРОС 1.2 допускает одновременную работу до четырех пользователей и 20 процессов. Достоинство ОС МИКРОС 1.2 — совместимость с широко распространенной мобильной ОС UNIXV6. В отличие от зарубежных и отечественных аналогов (ОС UNIX, ИНМОС, ДЕМОС, МНОС) ОС МИКРОС 1.2 может использоваться на мини- и микроЭВМ с малым объемом ОЗУ (около 64 Кбайт), не оснащенных аппаратурой диспетчера памяти (СМЗ, «Электроника 60», «Электроника 60М», ДВК). Это позволяет расширить область применения таких мини- и микроЭВМ, так как существующие для них ОС (РАФОС, ФОДОС, ОС ДВК), не являясь инструментальными, предоставляют пользователю весьма ограниченный набор средств разработки, отладки и сопровождения ПО. Высокая эффективность ОС МИКРОС 1.2 достигается за счет использования ее совместно с электронным диском.

Опыт использования показал следующее: применение ЭД для свопинга и размещения временных файлов совершенно оправданно. Время реакции системы на интерактивные запросы сократилось на порядок по сравнению с вариантом, базирующимся исключительно на гибких дисках;

размещение на ЭД утилит, хотя и существенно ускоряет работу, но требует его предварительной загрузки с гибких дисков (до 20 мин). Поэтому имеет смысл поступать так не со всеми утилитами, а лишь с наиболее необходимыми, например с проходами компилятора; размещение на ЭД рабочих файлов рискованно, так

как при сбоях питания информация теряется. Их более целесообразно держать на гибких дисках;

использование в накопителе корректирующего кода необходимо из-за невысокой надежности полупроводниковой памяти.

152140. Переславль-Залесский, ИПС АН СССР, лаборатория вычислительной техники

ЛИТЕРАТУРА

1. Злотник Е. М., Стежко И. К., Анищенко В. В., Киркоров С. И. «Электронный диск» для вычислительных комплексов на базе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 90—92.
2. Сорокин Ю. Ю., Лаврентьев В. В., Максимак С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60М» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 92.
3. Лукьянов Д. А. «Электроника 256К» — эмулятор диска для комплексов на основе микроЭВМ «Электроника 60» и ДВК // Микропроцессорные средства и системы. — 1986. — № 2. — С. 62—65.

Статья поступила 14.08.87

УДК 681.316

Д. А. Хохлов, М. Я. Масленков, М. Г. Крюков

ИСПОЛЬЗОВАНИЕ БИС КР580ГФ24 В ЗАДАЮЩЕМ ГЕНЕРАТОРЕ ДИСПЛЕЙНОГО МОДУЛЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Особенность современных дисплеев — микропроцессорное управление их работой. При этом данные от микропроцессора (МП) обычно передаются посредством прямого доступа к ОЗУ дисплея или используется программно-управляемая передача данных [1].

В первом режиме скорость обмена данными высока, однако при этом существенно увеличиваются аппаратные затраты [1, 2]. Второй режим требует больших затрат времени, но упрощает аппаратные средства. Это важно при построении дисплейных модулей для МП систем.

При построении дисплеев с программно-управляемой передачей данных одна из задач — обеспечить синхронность работы МП и синхрогенератора дисплея. При синхронной передаче данных доступ к ОЗУ дисплея обеспечивается путем регулярного, синхронного с ТВ-разверткой чередования адресов МП и синхрогенератора [1]. Но, учитывая некротность частот тактовых генераторов МП и синхрогенератора дисплея, а также то, что сигналы записи и чтения из ОЗУ формируются микропроцессором в моменты, когда с ОЗУ работает синхрогенератор, реализовать этот режим достаточно сложно. Применение общего тактового генератора также не даст синхронности, поскольку МП к ОЗУ дисплея должен обращаться в разрешенные моменты времени (определяемые синхрогенератором). Это приводит к наваливанию МП, усложняет логику его работы и снижает производительность [3].

Обеспечить синхронность работы МП и синхрогенератора с общим задающим генератором на БИС КР580ГФ24 можно при равенстве коэффициента деления частоты этой БИС числу точек в знаменателе отображаемого символа по горизонталю. Условно удовлетворяет формат отображаемого символа 7×9, а с учетом пробелов — 9×12 точек. Это позволяет за девять тактов обращаться к ОЗУ дисплея синхрогене-

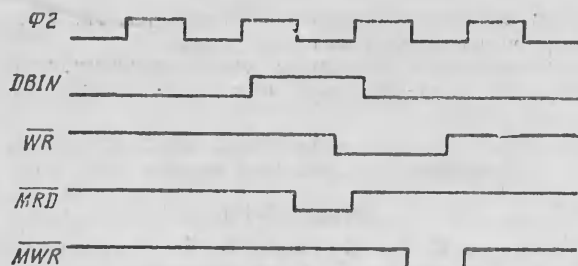


Рис. 1. Временная диаграмма формирования сигналов управления

ратору и, в случае необходимости, микропроцессору. Кроме того, данный формат дает возможность отображать прописные и строчные символы различных алфавитов и обладает лучшей различимостью на экране [4].

Авторами разработан дисплейный модуль для работы в составе МП систем на базе КР580ВМ50 с программно-управляемой передачей данных. Структура модуля — типовая [1], поэтому рассмотрим только особенности формирования сигналов управления работой ОЗУ и построения устройства.

На рис. 1 показано формирование сигналов чтения MRD и записи MWR в ОЗУ дисплея из соответствующих сигналов DBIN и WR [5] относительно тактового сигнала Ф2 с выхода БИС КР580ГФ24. При этом при низком уровне сигнала Ф2 разрешается работа МП с ОЗУ, а при высоком — синхрогенератора.

Принципиальная схема устройства формирования сигналов управления дисплеем приведена на рис. 2.

При высоком уровне сигнала Ф2 в сдвиговый регистр D4 проводится параллельная запись кода с выхода знакогенератора первым импульсом сигнала OSC. Следующие восемь импульсов сдвигают информацию в регистре D4, которая последовательным кодом поступает на сумматор, формирующий полный ТВ-сигнал. Сигнал параллельной записи в регистр D4 формируется элементом D3, на входы которого поступают прямой сигнал Ф2 и задержанный на 4 такта сдвиговым регистром D2 сигнал OSC. На последовательном входе элемента D4 присутствует Лог.0. Это обеспечивает фор-

мирование пробела между символами. Выходной сигнал элемента D3 — задающий для синхрогенератора (обеспечивает синхронность его работы и МП). Адреса ОЗУ дисплея переключаются на работу с МП или синхрогенератором сигналом Ф2 генератора D1.

Таким образом, при формате знакомета 9×12 точек БИС КР580ГФ24, тактируя работу синхрогенератора, позволяет упростить схему дисплейного модуля и избавиться от «мерцания» изображения. Небольшое снижение частоты работы тактового генератора с 18 до 15 МГц (что определяется временными параметрами развертки изображения) замедляет работу МП меньше, чем при использовании режима его приостановки. Кроме того, использование наиболее распространенного формата знакомета (9×12 точек) позволяет применять БИС знакогенератора типа К568РЕ1, К555РЕ4 [6].

Недостаток — малое время цикла дисплейного ОЗУ, что исключает использование БИС динамической и КМОП-памяти. Описанный способ построения синхрогенератора также неприменим при построении графических контроллеров, в которых число точек на «знакомето» кратно восьми.

В заключение отметим, что описанный способ формирования сигналов чтения (MRD) и записи (MWR) может быть использован не только при работе с ОЗУ дисплея, но и при построении различных устройств с МП управлением, в частности многомикропроцессорных систем, когда обращение одного МП к внешнему устройству должно осуществляться «незаметно» для другого управляющего устройства.

Телефон 567-10-18, Ленинград

ЛИТЕРАТУРА

1. Гуглин И. Н. Телевизионные устройства отображения информации. — М.: Радио и связь, 1981. — 200 с.
2. Панфилов Д. И. и др. Принципы организации и работы дисплеев на основе БИС КР580ВГ75 // Микропроцессорные средства и системы. — 1985. — № 3. — С. 51.
3. Дисплей СМ-7209, Техническое описание, 1984.
4. Безродный М. С. Классификация и характеристики дисплеев для микроЭВМ // Микропроцессорные средства и системы. — 1986. — № 4. — С. 28.
5. ОСТ 11.348.917—82. Микросхемы интегральные полупроводниковые серии КР580. Руководство по применению.
6. Назарьян А. Р. и др. Накопитель информации КР555РЕ4 емкостью 16 К для генератора символов // Электронная промышленность. — 1981. — № 2. — С. 46.

Статья поступила 29.01.88

УДК 681.323

А. П. Алешин, С. Н. Крюков

КОНТРОЛЛЕР КНМЛ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Контроллер КНМЛ позволяет использовать стандартное устройство СМ5211 для работы с микроЭВМ «Электроника 60» в операционной среде ОС РАФОС, RSX и т. д. Для пользователя контроллер представляет собой два программно-доступных регистра: регистр состояния (RS) с адресом 177500 и регистр данных (RD) с адресом 177502. Так как обмен с КНМЛ ведется байтами, то в RD используются только восемь младших разрядов. Форматы регистров приведены на рис. 1, форматы командного и информационного байтов КНМЛ СМ5211 — на рис. 2.

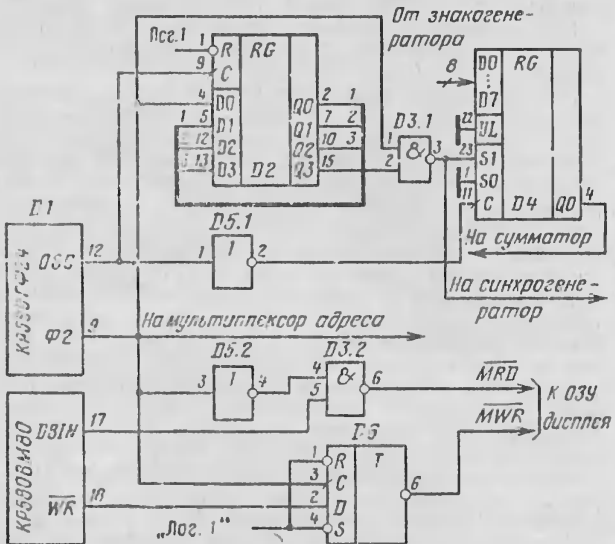


Рис. 2. Принципиальная схема устройства формирования сигналов управления:

- 1 — КР580ГФ24, 2 — К555ТМ8, 3 — К555ЛА3, 4 — К156ИР13, 5 — К555ЛН1, 6 — К555ТМ2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ошибка КНМЛ	Ошибка циклического контроля	Маркер ИЛ	Защита записи	Метка файла	Ошибка синхронизации	Не готов	Выбор накопителя	Запрос передачи	Разрешение прерывания	Свободен	Конец блока данных	Код команды			Пуск

а

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Не используются							байт данных								

б

Рис. 1. Форматы регистров состояний (а) и данных (б)

Функционально контроллер состоит из блоков интерфейса с магистралью микроЭВМ, прерываний, сопряжения с интерфейсом КНМЛ, формирования регистров состояния и данных. Блок интерфейса предназначен для приема и дешифрации адресов регистров, выработки сигналов управления и синхронизации обмена данными с магистралью.

В блоке прерывания вырабатываются сигнал КТНР Н и адрес вектора прерывания. Блок сопряжения с интерфейсом КНМЛ обеспечивает обмен данными

Д7-И	Д6-И	Д5-И	Д4-И	Д3-И	Д2-И	Д1-И	Д0-И	7	6	5	4	3	2	1	0
Маркер КНМЛ	Код команды						Ошибка циклического контроля	Маркер ИЛ	Защита записи	Метка файла	Ошибка синхронизации	Не готов	Свободен		

а

б

Рис. 2. Форматы командного (а) и информационного (б) байтов КНМЛ

в соответствии с протоколом ИРПР и разделение принятого байта данных и байта состояния. В блоке формирования регистров состояния и данных происходит преобразование форматов передаваемых данных, команд и состояния.

Работу контроллера (рис. 3) целесообразно рассмотреть на примере прохождения команд управления КНМЛ.

Команды: записи метки файла, возврата на блок, возврата на файл, пропуска файла, пропуска блока, прерывки

В адресной части канального цикла микроЭВМ адрес RS через примопередатчики D4...D7 дешифрируется элементами D10, D3, D1.1 и по фронту сигнала К СИАН записывается в триггер D9, вырабатывая син-

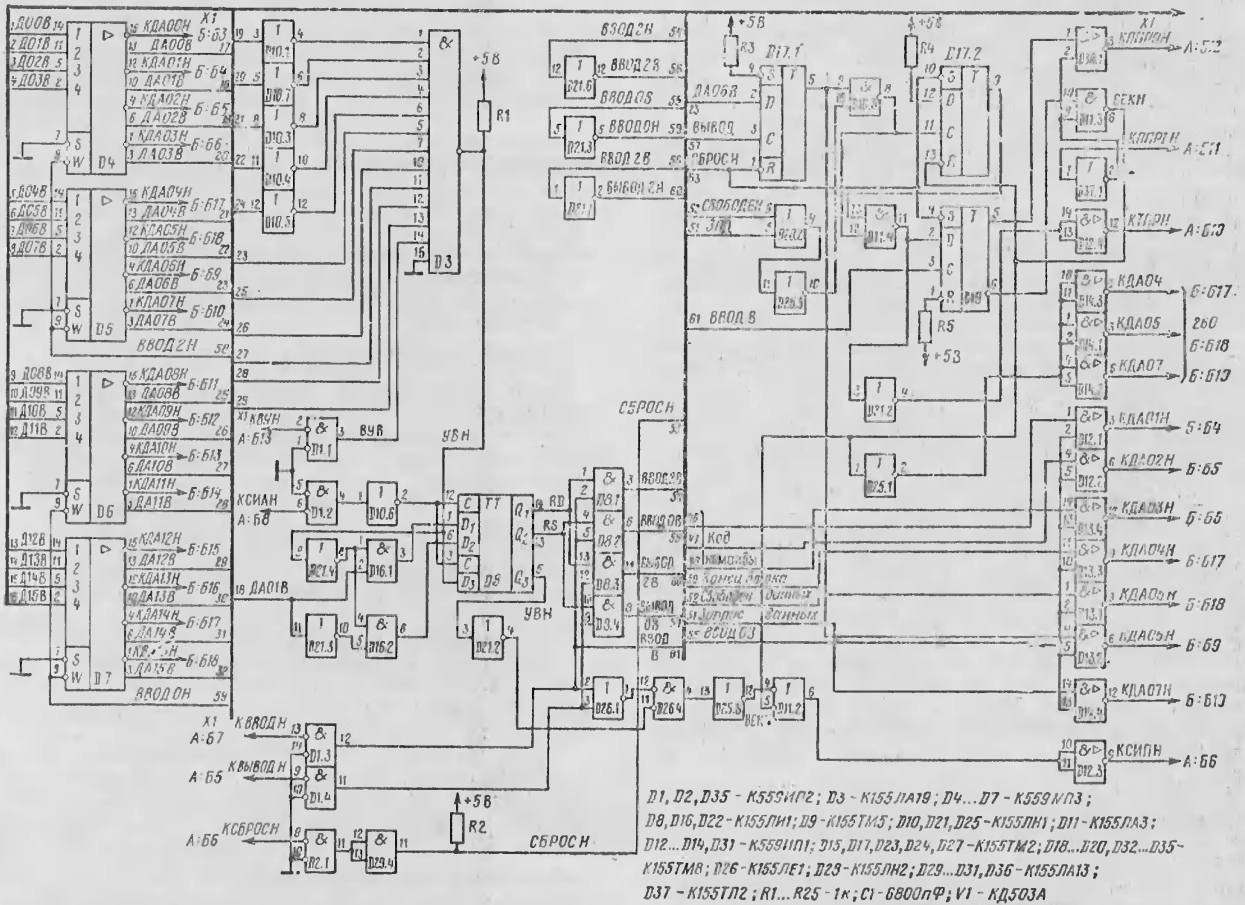


Рис. 3. (Начало)

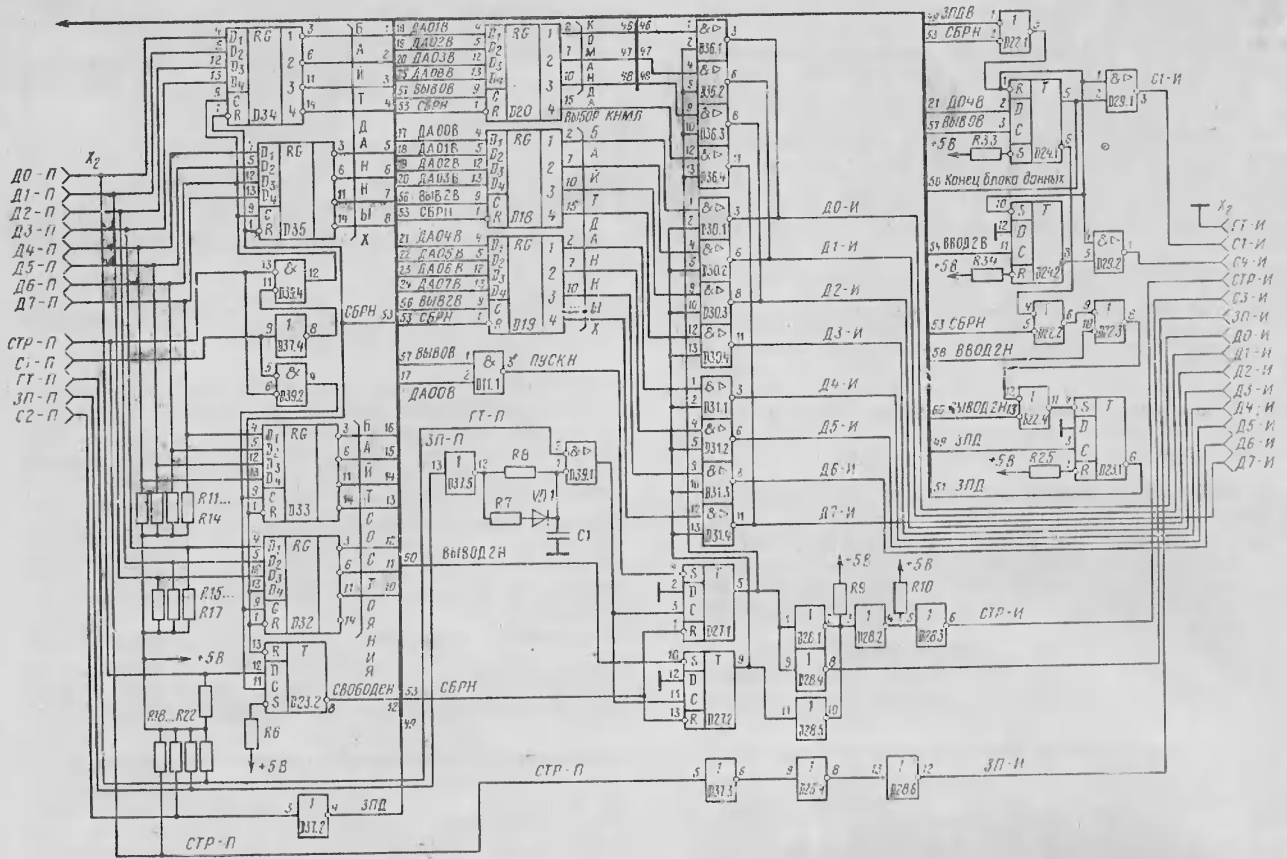


Рис. 3. Принципиальная схема контроллера (окончание)

нал «устройство выбрано». Этот же триггер служит для идентификации обращения к RS или RD по разряду ДА01Н адреса.

С приходом сигнала К ВЫВОД Н (или К ВВОД Н) в зависимости от состояния триггера D9 на выходе элементов D8 вырабатываются сигналы ВВОД 2 В — чтение регистра данных, ВВОД 0 В — чтение регистра состояния, ВЫВОД 2 В — запись в регистр данных, ВЫВОД 0 В — запись в регистр состояния.

Группа элементов D13, D14, D26.4, D25.6, D11.2... D12.3 вырабатывает сигналы К СИП Н. Байт команды, стробируемый сигналом ВЫВОД 0 В, записывается в регистр D20, а затем с помощью инверторов D36 приводится в соответствие с форматом команды КНМЛ CM5211 (см. рис. 2).

Сигнал ДА00В «Пуск» совместно с сигналом ВЫВОД 0 В через элемент D11.1 устанавливает триггер D27.1 в состояние Лог.1 и через элементы D28.1...D28.3 формирует сигнал СТ-И, поступающий на разъем X2. Элемент D28.4 выдает сигнал СЗ-И (идентификатора команды или байта данных), также поступающий на разъем X2. Байт команды, стробируемый сигналом с вывода 5 триггера D27.1, также поступает на разъем X2.

В соответствии с протоколом обмена сигнал СТ-И сбрасывает сигнал КНМЛ CM5211 ЗП-П, который через элементы D37.5, R7, R8, VD1, C1 (задержка) при наличии сигнала ГТ-П через элемент D39 сбрасывает триггер D27.1, тем самым прекращая действие сигнала СТ-И. Начинается выполнение команды.

По выполнению команды КНМЛ устанавливается байт состояния (см. рис. 2). Под действием сигналов СТ-П

и С1-П (низкого уровня) элемент D39.2 стробирует прохождение байта состояния КНМЛ в регистры D33, D32 и D23.2. Из этих регистров он может быть прочитан в ЭВМ (см. рис. 1).

Команда записи блока данных

После формирования байта команды и начала выполнения команды КНМЛ выставляет сигнал С2-И (запрос передачи данных). Этот сигнал через инвертор D37.2 запоминается триггером D23.1, откуда поступает через передатчик D14.4 в канал ЭВМ в виде седьмого разряда RS. ЭВМ после считывания данных из KS и анализа седьмого разряда формирует байт данных, который записывается в регистры D18, D19. Сигнал ВЫВОД 2 Н (элемент D21.1) обеспечивает прохождение байта данных на разъем X2 и установление сигнала СТ-И для сброса сигнала ЗП-П, т. е. перевода его в состояние высокого уровня.

Элемент D28.4 формирует сигнал опознавания байта данных СЗ-И. При записи байта данных сигнал ВЫВОД 2 Н сбрасывает (по выходу 6) триггер D23.1, прекращая тем самым действие сигнала С2-П.

Таким образом происходит побайтный ввод данных в КНМЛ до тех пор, пока на очередной запрос передачи данных не будет записана в четвертый разряд RS Лог.1 — конец блока данных. Этот сигнал через триггер D24.1 формирует интерфейсный сигнал С1-И.

Команда чтения блока данных

После запуска команды КНМЛ устанавливает на линиях Д0-П...Д7-П байт данных, который заносится в регистры D34, D35, стробируемые сигналом СТ-П.

Сигнал С2-П, так же как и при записи, генерирует единицу в седьмом разряде RS. Данные в ЭВМ считываются через приемопередатчики D4, D5 под действием сигнала ВВОД 2 Н. Этот же сигнал сбрасывает через элементы D22.3, D22.4 триггер D23.1, прекращая тем самым действие С2-П.

Элементы D37.3, D25.4, D28.6 обеспечивают временное согласование сигналов СТР-П и ЗП-И. Сигнал ВВОД 2 Н сбросит триггер D24.2, с выхода 8 которого снимается сигнал С4-И (подтверждение приема данных). В ответ на сигнал С4-И через некоторое время КНМЛ снимает сигнал С2-П.

Работа контроллера в режиме прерываний

Для работы контроллера в режиме прерываний необходимо в разряд 6 RS записать Лог.1. При этом триггер D17.1 также устанавливается в состояние Лог.1. Прерывания происходят в одном из двух случаев:

по окончании выполнения очередной команды КНМЛ при выработке сигнала «свободен»;

в режиме байтового обмена данными с КНМЛ по сигналу С2-П.

Появление одного из этих сигналов через элементы D26.2, D25.5, D16.3, D17.2, D11.4, D21.2, D12.4 инициирует каналный сигнал К ТПР Н. Триггер D15.1 и элемент D38.1 запрещают распространение сигнала К ППРО Н. По сигналу ВВОД В через элементы D14.1, D14.2, D14.3 адрес вектора прерываний 260 передается в канал ЭВМ.

Операционная система MICROS

Операционная система MICROS поддерживает вычислительную систему, включающую микроЭВМ «Электроника 60М», КНМЛ СМ5211.09, цифровое печатающее устройство DZM-180, видеотерминал ВТА-2000/10, фотосчитыватель с перфоленты FS-1501, перфоратор ПЛ-150.

Она позволяет использовать программы, хранящиеся на перфоленте и разработанные с помощью перфоленточной ОС копировать файлы, записанные на МЛ в символьном или двоичном формате, с возможностью получения копий на перфоленте и текстовых файлов на DZM, загружать программы, записанные на кассету в абсолютном двоичном формате, в ОЗУ ЭВМ и запускать их на счет, если это необходимо, редактировать файлы с текстами программ.

ОС MICROS состоит из управляющей и вычислительной частей. Ядро управляющей части составляет монитор, который в процессе автозагрузки системы с КНМЛ размещается в ОЗУ ЭВМ.

Монитор обеспечивает загрузку и перезагрузку в ОЗУ системных компонентов MICROS, поддерживает диа-

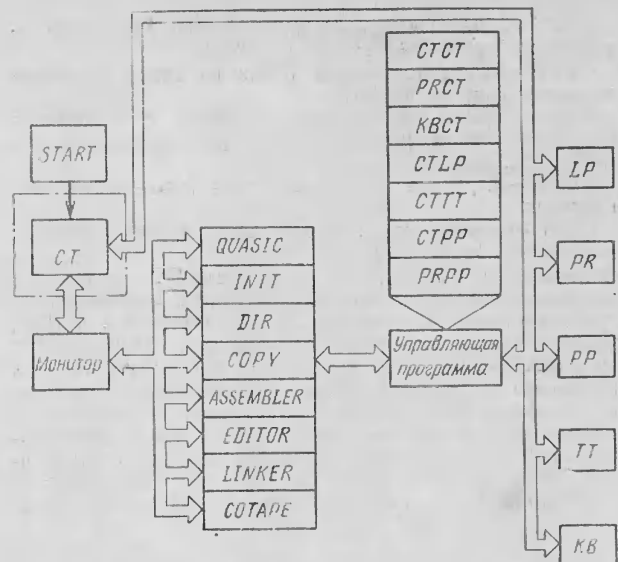


Рис. 4. Взаимодействие отдельных частей ОС

лог с пользователем и обмен ЭВМ с системными устройствами, контролирует работу и диагностирует ошибки КНМЛ, осуществляет синтаксический контроль вводных команд, осуществляемых пользователем, а также принудительный (по желанию пользователя) и автоматический (при наличии ошибок) режимы перезапуска системы.

В качестве вычислительной части ОС используется адаптированная к MICROS система QUASIC*. В последнюю включен ряд команд, с помощью которых текст исходной программы вводится с кассеты, а текст отредактированной программы выводится на кассету.

ОС MICROS включает в себя следующие программы: INIT — начальная инициализация МЛ. Делает МЛ пригодной для использования в системе;

DIR — распечатка каталога ленты;

COPY — обмен файлами между системными устройствами;

COTAPE — перезапись с кассеты на кассету с запросами на подтверждение того или иного файла.

Обмен информацией между системными устройствами обеспечивается набором управляющих программ:

CTCT — передача файлов с одной кассеты КНМЛ на другую;

* Подольский Л. И., Лясковский А. П. Система программирования QUASIC-2 для микроЭВМ // Микропроцессорные средства и системы. — 1987. — № 2. — С. 9—11.

Именной блок

Длина блока 1 байт	Имя файла 10 байт max	Число строк 2 байта	1	2	...	Последний
--------------------	-----------------------	---------------------	---	---	-----	-----------

Строка данных

Длина строки 1 байт	байты кодов ASCII
---------------------	-------------------

Рис. 5. Формат символьного файла

Именной блок

Длина блока 1 байт	Имя файла 10 байт max	1	2	...	Последний
--------------------	-----------------------	---	---	-----	-----------

Блок данных

Длина блока 2 байта	Адрес загрузки 2 байта	байты двоичных данных	DKC
---------------------	------------------------	-----------------------	-----

Рис. 6. Формат двоичного файла (DKC — дополнение до контрольной суммы)

PRCT — передача файлов с устройства PR (фотосчитыватель) на устройство СТ (КНМЛ);

СТП — вывод текстового файла на экран видеотерминала с кассеты КНМЛ;

СТЛР — распечатка текстового файла с устройства СТ (КНМЛ) на устройство LP (устройство печати DZM-180);

PRPP — передачи файлов с фотосчитывателя на перфоратор;

СТРР — перфорации символьных и двоичных файлов, записанных на кассету.

Начальная автозагрузка и запуск ОС происходят автоматически при включении питания с помощью программы начальной загрузки START, занесенной в ПЗУ, реализованное на четырех микросхемах K556PT4. ОС занимает минимальный объем ОЗУ микроЭВМ, так как постоянно в памяти содержится программа монитор, а остальные компоненты загружаются с кассеты по мере надобности и взаимодействуют при каждом последующем обращении к очередной системной программе. Объем ядра ОС MICROS составляет 1038 слов.

Взаимодействие отдельных частей ОС можно пред-

ставить в виде блок-схемы (рис. 4), где LP — устройство мозаичной печати, PR — фотосчитыватель, PP — перфоратор, П — экран терминала, KB — клавиатура.

ОС оперирует с символьными (текстовыми) и двоичными файлами. В символьных файлах каждая строка текста записывается в отдельные блоки данных (рис. 5). В именном блоке в двух последних байтах записывается число блоков-строк текста в файле. В двоичных файлах формат блоков данных совпадает с форматом абсолютной ленты, за исключением двух первых байтов: 001 — начало блока и 000 — нулевая строка, присутствующих на абсолютных лентах и отсутствующих на МЛ (рис. 6). Это различие учитывается управляющими программами PRCT и СТРР.

В настоящее время адаптируются в качестве системных компонентов программы ассемблера, редактора текстов и компоновщика.

Телефон 516-99-29. Ленинград, Алёшин Александр Николаевич (звонить после 19.00)

Статья поступила 03.09.87

УДК 681.324

В. Н. Фонов, И. Л. Костромин, Б. А. Калинин

БЛОК СВЯЗИ МИКРОЭВМ «ЭЛЕКТРОНИКА 60М» с КНМЛ EC5091

Блок связи предназначен для байтового параллельного обмена с каналом данных-адреса микроЭВМ в режимах опроса флага и прерывания и последовательного обмена с двумя КНМЛ. Преобразование параллельного кода в последовательный и наоборот, контроль четности и кодирование передаваемой и принимаемой в последовательном коде информации осуществляет универсальное синхронно-асинхронное программируемое приемопередающее устройство (УСАПП) K580IK51.

Блок выполнен на плате размером 135×240 мм, имеет концевые печатные контакты для соединения с монтажной панелью микроЭВМ «Электроника 60М». Подсоединение КНМЛ к блоку связи осуществляется через разъем МРН-32-1.

Внутри блока связи организована 8-разрядная двунаправленная магистраль, по которой происходит обмен данными с УСАПП (адрес 177556), программирование в течение состояния УСАПП (адрес 177552), программное управление КНМЛ и чтение состояния блока связи и КНМЛ (адрес 177554). Восемьразрядный регистр команд обеспечивает разрешение прерывания (6-й разряд); выбор КНМЛ (2-й и 3-й разряды); управление движением ленты (0-й разряд — вперед, 1-й — назад, 7-й — ускоренное движение), работу КНМЛ в режимах записи (4-й разряд) и воспроизведения (5-й разряд). В ре-

ПРОГРАММА	ВВОДА В
РЕСОЛЮТНОМ	ФОРМАТЕ

107424	010706	MOV	Z7,Z6	107604	012701	MOV	#177554,Z1
107426	062706	RDD	#177510,Z6	107606	177554		
107430	177510			107610	105711	TSTB	(Z1)
107432	010705	MOV	Z7,Z5	107612	100376	RPL	107610
107434	062705	RDD	#000130,Z5	107614	016101	MOV	000002(Z1),Z1
107436	000130			107616	000002		
107440	012700	MOV	#012045,Z0	107620	060100	RDD	Z1,Z0
107442	012045			107622	005302	DEC	Z2
107444	012701	MOV	#120240,Z1	107624	000207	RTS	Z7
107446	120240			107626	004765	JSR	Z7,000020(Z5)
107450	004767	JSR	Z7,107666	107630	000020		
107452	000212			107632	005301	DEC	Z1
107454	016703	MOV	107772,Z3	107634	001374	BNE	107626
107456	000312			107636	004765	JSR	Z7,000020(Z5)
107460	004765	JSR	Z7,000042(Z5)	107640	000020		
107462	000042			107642	005701	TST	Z1
107464	022704	CMP	#000006,Z4	107644	001370	BNE	107626
107466	000006			107646	005000	CLR	Z0
107470	001411	BEQ	107514	107650	004715	JSR	Z7,(Z5)
107472	004715	JSR	Z7,(Z5)	107652	010402	MOV	Z4,Z2
107474	060403	RDD	Z4,Z3	107654	162702	SUB	#000004,Z2
107476	004715	JSR	Z7,(Z5)	107656	000004		
107500	010423	MOV	Z4,(Z3)+	107660	000207	RTS	Z7
107502	005702	TST	Z2	107662	000000	HALT	
107504	001374	BNE	107476	107664	000000	HALT	
107506	004767	JSR	Z7,107546	107666	110167	MOV	Z1,107712
107510	000034			107670	000020		
107512	000760	BR	107454	107572	000301	SWAB	Z1
107514	004715	JSR	Z7,(Z5)	107674	042701	BIC	#177400,Z1
107516	004767	JSR	Z7,107546	107676	177400		
107520	000024			107700	110167	MOV	Z1,107740
107522	005000	CLR	Z0	107702	000034		
107524	012701	MOV	#000207,Z1	107704	010037	MOV	Z0,0#177554
107526	000207			107706	177554		
107530	004767	JSR	Z7,107666	107710	000005	RES	
107532	000132			107712	000207	RTS	Z7
107534	006004	ROR	Z4	107714	004767	JSR	Z7,107760
107536	103402	BCS	107544	107716	000040		
107540	000304	ASL	Z4	107720	012737	MOV	#000376,0#177552
107542	000114	JMP	(Z4)	107722	000376		
107544	000000	HALT		107724	177552		
107546	004765	JSR	Z7,000020(Z5)	107726	000300	SWAB	Z0
107550	000020			107730	004767	JSR	Z7,107760
107552	105700	TSTB	Z0	107732	000034		
107554	001001	BNE	107560	107734	010037	MOV	Z0,0#177552
107556	000007	RTS	Z7	107736	177552		
107560	020626	CMP	Z6,(Z6)+	107740	000000	HALT	
107562	000716	BR	107420	107742	004767	JSR	Z7,107760
107564	004767	JSR	Z7,107604	107744	000012		
107566	000014			107746	013701	MOV	0#177552,Z1
107570	010104	MOV	Z1,Z4	107750	177552		
107572	004767	JSR	Z7,107604	107752	030127	BIT	Z1,#020040
107574	000006			107754	000040		
107576	000301	SWAB	Z1	107756	001754	BEQ	107710
107600	050104	BIS	Z1,Z4	107760	012701	MOV	#020400,Z1
107602	000207	RTS	Z7	107762	000400		
				107764	005301	DEC	Z1
				107766	001376	BNE	107764
				107770	000207	RTS	Z7

Рис. 1


```

107140 010706 MOV 27,26
107142 062706 ADD #177774,26
107144 177774
107146 010703 MOV 27,23
107150 062703 ADD #000134,23
107152 000134
107154 012700 MOV #000425,20
107156 000125
107160 012701 MOV #103640,21
107162 103640
107164 004767 JSR 27,107766
107166 000576
107170 005000 CLR 20
107172 012704 MOV #000100,24
107174 000100
107176 004713 JSR 27,(23)
107200 005304 DEC 24
107202 001375 BNE 107176
107204 016701 MOV 107776,21
107206 000566
107210 010102 MOV 21,22
107212 052702 ADD #000100,22
107214 000100
107216 020267 CMP 22,107774
107220 000552
107222 103003 BCC 107232
107224 004767 JSR 27,107270
107226 000040
107230 000770 BR 107212
107232 016702 MOV 107774,22
107234 000536
107236 004767 JSR 27,107270
107240 000026
107242 016701 MOV 107772,21
107244 000524
107246 005002 CLR 22
107250 004767 JSR 27,107270
107252 000014
107254 012701 MOV #000207,21
107256 000207
107260 004767 JSR 27,107766
107262 000502
107264 000000 HALT
107266 000740 BR 107170
107270 005000 CLR 20
107272 005200 INC 20
107274 004713 JSR 27,(23)
107276 004763 JSR 27,000030(23)
107300 000030
107302 000207 RTS 27

```

```

107304 004717 JSR 27,(27)
107306 010146 MOV 21,(26)
107310 012701 MOV #177554,21
107312 177554
107314 105711 TSTB (21)
107316 100376 BPL 107314
107320 010061 MOV 20,000002(21)
107322 000002
107324 050005 ADD 20,25
107326 000300 SWAB 20
107330 012601 MOV (26)+,21
107332 000207 RTS 27
107334 005005 CLR 25
107336 062700 ADD #000005,20
107340 000005
107342 020102 CMP 21,22
107344 101023 BHI 107414
107346 010204 MOV 22,24
107350 160104 SUB 21,24
107352 060400 ADD 24,20
107354 004713 JSR 27,(23)
107356 010100 MOV 21,20
107360 004713 JSR 27,(23)
107362 112100 MOVB (21)+,20
107364 004763 JSR 27,000002(23)
107366 000002
107370 005304 DEC 24
107372 001373 BNE 107362
107374 005400 NEC 25
107376 110500 MOVB 25,20
107400 042700 BIC #177400,20
107402 177400
107404 004713 JSR 27,(23)
107406 005000 CLR 20
107410 004713 JSR 27,(23)
107412 000747 BR 107332
107414 004713 JSR 27,(23)
107416 010100 MOV 21,20
107420 004713 JSR 27,(23)
107422 000764 BR 107374

```

Рис. 2

```

000000 012711 MOV #000045,(21)
000002 000045
000004 000005 RES
000006 012737 MOV #000376,C#177552
000010 000376
000012 177552
000014 012700 MOV #000400,20
000016 000400
000020 005300 DEC 20
000022 001376 BNE 000020
000024 012737 MOV #000024,C#177552
000026 000024
000030 177552
000032 012700 MOV #000400,20
000034 000400
000036 005300 DEC 20
000038 001376 BNE 000036
000042 013703 MOV C#177552,23
000044 177552
000046 030327 BIT 23,#000040
000050 000040
000052 001754 BEQ 000004
000054 004715 JSR 27,(25)
000056 001776 BEQ 000054
000060 022700 CMP #000351,20
000062 000351
000064 001373 BNE 000054
000066 004715 JSR 27,(25)
000070 005003 CLR 23
000072 004715 JSR 27,(25)
000074 110022 MOVB 20,(22)+
000076 005304 DEC 24
000100 001374 BNE 000072
000102 004715 JSR 27,(25)
000104 010411 MOV 24,(21)
000106 000000 HALT
000110 105711 TSTB (21)
000112 100376 BPL 000110
000114 016100 MOV 000002(21),23
000116 000002
000120 050003 ADD 20,23
000122 000207 RTS 27

```

R1=177554
R2=(AAPE3 ЗАГРУЗКИ)
R4=315
R5=110
R6=(AAPE3 ЗАГРУЗКИ)-2

Рис. 3.

гистре состояния блока связи 7-й разряд — готовности (формируется из сигналов готовности приемника или передатчика УСАПП и готовности КНМЛ); 6-й — разрешение требования прерывания; 5-й — нет защиты записи; 0-й — начало-конец ленты.

В блоке связи предусмотрен кварцевый генератор на 2 МГц для синхронизации работы УСАПП. Возможность работать со стандартными устройствами ввода-вывода, не вынимая плату блока связи, обеспечивает переключатель в дешифраторе адреса, запрещающий работу последнего. Переключатель размещен на внешнем разъеме платы блока.

Обмен с КНМЛ ведется со скоростью 4 Кбит/с. При плотности записи 32 бит/мм стандартная кассета имеет емкость около 0,5 Мбайт.

Обмен микроЭВМ с КНМЛ можно вести в абсолютном формате, используя блоки, аналогичные програм-

мам «абсолютный загрузчик» и «перфорация в абсолютном формате». Обе программы скомпонованы в перемещаемый модуль и работают с общей программой управления КНМЛ. Регистром перемещения для программы ввода в абсолютном формате (рис. 1) служит ячейка ОЗУ 107772. Стартовый адрес программы 107424. Программа вывода в абсолютном формате (рис. 2) разбивает заданный файл на блоки и выводит на магнитную ленту. Адреса начала и конца файла помещают соответственно в ячейки 107776 и 107774, а четный адрес передачи управления (или 1) — в 107772. Стартовый адрес 107140.

Программный блок ввода-вывода в абсолютном формате размещается в начале магнитной ленты (в начальном формате). Программа загрузки в начальном формате (рис. 3) вводится с клавиатуры или из ПЗУ. В последнем случае ее необходимо до-

поднять командами загрузки указательных регистров. При правильной загрузке в ОЗУ программного блока ввода-вывода младший байт R3=0 (рис. 4).

Для работы блока используется компилирующая система программирования QUASIC-2* версии V2.10, содержащая операторы обмена с перфоленитой, модифицированные для работы с КНМЛ. В списке адресов регистров состояния компилятора адрес фотосчитывателя 177550 заменяется на адрес блока связи 177554. В версии V2.10 этот адрес находится в ячейке 10000 ОЗУ. Далее в тексте компилятора программные блоки взаимодействия с устройствами ввода-вывода операторов SAVE и OLD

* Подольский Л. И., Ляскоцкий А. П. Система программирования QUASIC-2 для микроЭВМ // Микропроцессорные средства и системы. — 1987. — № 2. — С. 9.—11.

```
001000 012706 MOV #107136, Z6
001002 107136
001004 012703 MOV #107304, Z3
001006 107304
001010 012700 MOV #000425, Z0
001012 000425
001014 012701 MOV #103640, Z1
001016 103640
001020 004737 JSR Z7, @#107666
001022 107666
001024 005000 CLR Z0
001026 012701 MOV #000100, Z1
001030 000100
001032 004713 JSR Z7, (Z3)
001034 005301 DEC Z1
001036 001375 BNE 001032
001040 012700 MOV #000351, Z0
001042 000351
001044 004713 JSR Z7, (Z3)
001046 012700 MOV #107140, Z0
001050 107140
001052 012702 MOV #107772, Z2
001054 107772
001056 012704 MOV #000315, Z4
001060 000315
001062 005005 CLR Z5
001064 004763 JSR Z7, 000056 (Z3)
001066 000056
001070 005000 CLR Z0
001072 012701 MOV #000207, Z1
001074 000207
001076 004737 JSR Z7, @#107666
001100 107666
001102 000000 HALT
001104 000735 ER 001000
```

Рис. 4

```
040172 022701 CMP #177554, Z1
040174 177554
040176 001404 BEQ 040210
040200 105711 TSTB (Z1)
040202 100376 BPL 040200
040204 000137 JMP @#025104
040206 025104
040210 012711 MOV #000025, (Z1)
040212 000025
040214 000005 RES
040216 012767 MOV #000400, 040226
040220 000400
040222 000002
040224 005327 DEC #000000
040226 000000
040230 001375 BNE 040224
040232 012737 MOV #000376, @#177552
040234 000376
040236 177552
040240 012767 MOV #000400, 040250
040242 000400
040244 000002
040246 005327 DEC #000000
040250 000000
040252 001375 BNE 040246
040254 012737 MOV #000001, @#177552
040256 000001
040260 177552
040262 062737 ADD #000076, 0#025102
040264 000076
040266 025102
040270 105711 TSTB (Z1)
040272 100376 BPL 040270
040274 112267 MOVB (Z2)+, 040302
040276 000002
040300 012761 MOV #000000, 000002 (Z1)
040302 000000
040304 000002
040306 022767 CMP #000012, 040302
040310 000012
040312 177666
040314 001440 BEQ 040416
040316 022767 CMP #000014, 040302
040320 000014
040322 177756
040324 001050 BNE 040446
040326 012767 MOV #000002, 040440
040330 000002
040332 000104
```

Рис. 5

```
040334 012767 MOV #000737, 040444
040336 000737
040340 000102
040342 000430 BR 040424
040344 000005 RES
040346 005011 CLR (Z1)
040350 012737 MOV #000100, @#177560
040352 000100
040354 177560
040356 062737 ADD #000004, @#025102
040360 000004
040362 025102
040364 012767 MOV #000456, 040300
040366 000456
040370 177706
040372 012767 MOV #000240, 040440
040374 000240
040376 000040
040400 012767 MOV #001403, 040442
040402 001403
040404 000034
040406 012767 MOV #000240, 040444
040410 000240
040412 000030
040414 000414 BR 040446
040416 012767 MOV #000010, 040440
040420 000010
040422 000014
040424 105711 TSTB (Z1)
040426 100376 BPL 040424
040430 012761 MOV #000000, 000002 (Z1)
040432 000000
040434 000002
040436 005327 DEC #000000
040440 000000
040442 001370 BNE 040424
040444 000240 NOP
040446 000137 JMP @#025110
040450 025110
040452 162737 SUB #000102, @#025102
040454 000102
040456 025102
040460 012767 MOV #012761, 040300
040462 012761
040464 177612
040466 012767 MOV #001370, 040442
040470 001370
040472 177746
040474 000764 BR 040446
```

замениются на обращения к соответствующим программам-драйверам (рис. 5--7). Для оператора SAVE:

адреса	25100/105711	замениются на	25100/000137
	25102/100376		25102/157272
для оператора OLD:	24710/005210	замениются на	24710/000240
	24712/032711		24712/000240
	24714/100200		24714/000137
	24716/001775		24716/157100
	24720/001412		24720/000240
	24766/016104		24766/000240
	24770/000002		24770/000240

БЛОК ПЕРЕМЕЩЕНИЯ

```
040550 012700 MOV #040000, Z0
040552 040000
040554 012701 MOV #040540, Z1
040556 040540
040560 012702 MOV #157100, Z2
040562 157100
040564 010103 MOV Z1, Z3
040566 160003 SUB Z0, Z3
040570 060302 ADD Z3, Z2
040572 062701 ADD #000002, Z1
040574 000002
040576 062702 ADD #000002, Z2
040600 000002
040602 014142 MOV -(Z1), -(Z2)
040604 020001 CMP Z0, Z1
040606 001375 BNE 040602
040610 000137 JMP @#001000
040612 001000
```

Обнаружить указанные места в тексте компилятора несложно, достаточно отключить устройство ввода-вывода, ввести нужный оператор и перевести клавишу «программа/пульт» «Электроника 60» в положение «пульт».

Для размещения программ-драйверов вне зоны ОЗУ, доступной компилятору, оператором SET устанавливается значение MEMTOP=157100.

Компилятор (два банка памяти ОЗУ) компонуется вместе с программами-драйверами и блоком перемещения в загрузочный файл. В текстах программ-драйверов и блока перемещения адреса соответствуют загрузочному файлу. По окончании ввода загрузочного файла с КИМЛ в ОЗУ (ячейка 107772=0) программа ввода в абсолютном формате передаст управление блоку перемещения,

Рис. 6

ЭЛЕКТРОННЫЙ ДИСК НА ОСНОВЕ ПЗУ

ПРОГРАММА-ДРАЙВЕР
ДЛЯ ОПЕРАТОРА OLD

```

040000 012711 MOV #000045,(Z1)
040002 000045
040004 000005 RES
040006 012737 MOV #000376,(R177552)
040010 000376
040012 177552
040014 012767 MOV #000400,040024
040016 020400
040020 000002
040022 005527 DEC #000020
040024 000000
040026 001375 BNE 040002
040030 012737 MOV #000024,(R177552)
040032 000004
040034 177552
040036 012767 MOV #000400,040040
040040 000100
040042 000002
040044 000327 DEC #000020
040046 000000
040050 001375 BNE 040044
040052 013767 MOV #0177552,040062
040054 177552
040056 000002
040060 032727 BIT #000017,#000040
040062 000247
040064 000040
040066 001745 BEQ 040004
040070 100711 TSTB (Z1)
040072 100376 BPL 040070
040074 000004 CLR Z4
040076 002737 ADD #000110,(R024711)
040080 000110
040082 024714
040084 000137 JMP 04024716
040086 024716
040088 100711 TSTB (Z1)
040090 100376 BPL 040110
040092 016100 MOV 000002(Z1),Z4
040094 000002
040096 020427 CMP Z4,#000014
040098 000014
040100 001011 BNE 040150
040102 000005 RES
040104 005511 CLR (Z1)
040106 012737 MOV #000005,(R177552)
040108 000100
040110 177552
040112 162737 SUB #000110,(R024711)
040114 000110
040116 024714
040118 000755 BR 040104
040120 032737 BIT #000070,(R177552)
040122 000070
040124 177552
040126 001752 BEQ 040104
040128 000004 CLR Z4
040130 012737 MOV #000024,(R177552)
040132 000024
040134 177552
040136 000024 BR 040104
040138 000745 BR 040104

```

Рис. 7

который разместит программы-драйверы в зоне ОЗУ 157100...157574 и передаст управление компилятору. Обмен с КНМЛ под управлением компилятора осуществляется операторами OLD, APP PR; и SAVE. Для работы программ необходима микроЭВМ с полным объемом памяти — 56 Кбайт. Использование программ-драйверов в другой конфигурации ОЗУ требует их адаптации.

Телефон 44-87-15, Свердловск

Статья поступила 27.10.86

Использование электронных дисков в качестве системных носителей резко увеличивает быстродействие системного устройства, применение ПЗУ в таком качестве повышает их надежность.

Однако непосредственное использование ПЗУ системного носителя для таких широко распространенных ОС, как RT-11, и ей подобных невозможно, так как системы в процессе работы могут модифицировать собственные компоненты. Кроме того, эти ОС используют для работы область свопинга системы SWAP.SYS, которая должна находиться на системном носителе.

Существует большое число универсальных ОС, поэтому нерационально создавать специализированные, способные работать в ПЗУ, так как они имеют весьма ограниченные возможности [1] и направлены на решение конкретных задач. При этом трудоемкость разработки и отладки таких ОС довольно высока.

Известно несколько вариантов реализации ЭД. По структуре их можно разделить на две группы:

ОС хранится на магнитном носителе. ЭД—ОЗУ большого объема (до 2 Мбайт) с файловой структурой [2, 3, 5]. ОС сначала загружается с магнитного носителя, затем копируется в ОЗУ и перезагружается с ЭД. При достаточно большом объеме ОЗУ на ЭД могут храниться файлы, необходимые пользователю для работы (языковые процессоры, разрабатываемые программы и др.). Подобные ЭД имеют большое быстродействие, значительно уменьшают

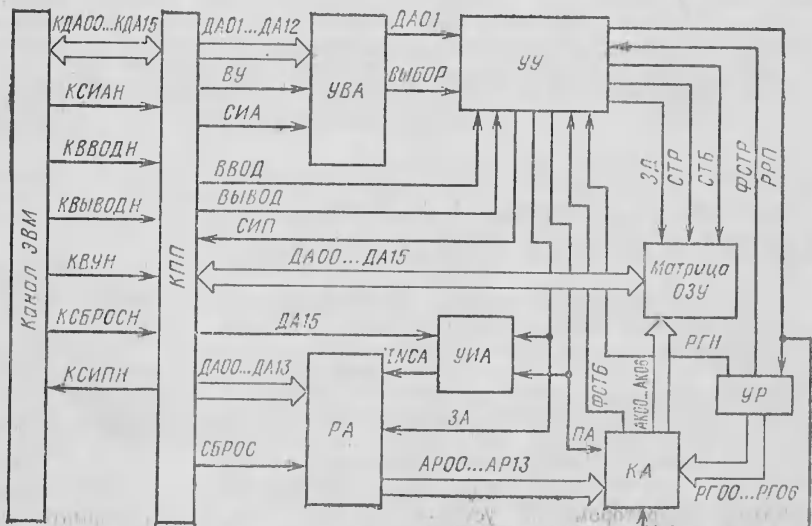
время работы с магнитным носителем и увеличивают срок его службы, по их надежности, особенно на этапе начальной загрузки, ограничивается надежностью хранения информации на магнитном носителе.

ОС хранится в ПЗУ. ЭД состоит из ПЗУ и ОЗУ. ОС загружается непосредственно из ПЗУ, в ОЗУ располагается файл SWAP.SYS и разрабатываемые программы пользователя [4]. Основное преимущество разработки — нет необходимости записывать ОС на магнитный носитель, резко возрастает надежность системного устройства. Такой ЭД может работать в запыленных средах, с высоким уровнем магнитных полей.

К сожалению, ЭД в настоящее время серийно не выпускаются. Повторение конструкций ЭД [3...6] затруднено, так как необходимо выполнить большой объем работ, связанных с изготовлением и настройкой устройства, а в ряде случаев требуется модернизация самой ЭВМ [3].

Была поставлена задача разработать надежный и компактный ЭД, максимально используя серийные модули микроЭВМ «Электроника 60».

В качестве ПЗУ применяется модуль МС3405, выпускаемый промышленностью. Его основные характеристики: емкость модуля при использовании микросхем К573РФ4 95 Кбайт, К573РФ41 и К573РФ42 48 Кбайт. В субблоке микроЭВМ можно одновременно применять до восьми модулей (максимальная суммарная емкость ПЗУ 768 Кбайт, 1536 блоков по 256 слов).



Структурная схема ОЗУ.

Для файла SWAP.SYS в качестве носителя выбрано ОЗУ (16 Кслов) с последовательным доступом (см. рисунок).

Для непосредственного размещения файла SWAP.SYS необходимо 6,5 Кслов (26 блоков), остальной объем — для хранения модифицируемых компонентов ОС.

Устройство выделения адреса (УВА) служит для начальной дешифрации адреса и запоминания состояния разряда ДАО1 в адресной части цикла обращения.

Устройство управления (УУ) осуществляет окончательную дешифрацию адреса, формирует сигналы: СИП, запись данных в ОЗУ (ЗД), стробирование адреса строки ячейки ОЗУ (СТБ), стробирование адреса столбца ячейки ОЗУ (СТР), сигнал разрешения регенерации памяти (РРП), запись адреса ячейки ОЗУ в регистр адреса (ЗА), переключение второй части коммутатора адреса ячейки ОЗУ (ПА). Регенерация памяти разрешается, если отсутствует обращение к регистру данных (в этом случае сигнал СИП запрещен).

Матрица ОЗУ состоит из 16 микросхем динамической памяти К581РУ4 (К565РУ3) (16К 16-разрядных слов).

Регистр адреса ОЗУ (РА) предназначен для записи адреса ячейки ОЗУ по сигналу ЗА, его хранения и увеличения адреса ячейки ОЗУ на единицу по сигналу INCA.

Устройство инкрементирования адреса ячейки ОЗУ (УИА) формирует данные сигнала INCA после окончания цикла ввода-вывода, если при обращении к регистру адреса был установлен разряд ДАО15. Устройство управляется сигналами ЗА и ПА, по сигналу ЗА запоминается содержимое разряда ДАО15.

Устройство регенерации (УР) осуществляет адаптивную регенерацию памяти. Если обращение к регистру данных происходит в момент регенерации памяти, сигнал РРП снимается, регенерация прекращается и вырабатывается сигнал СИП. Регенерация осуществляется по строкам, поэтому сигнал СТБ не вырабатывает-

ся. По сигналу ФСТР формируется очередной адрес регенерируемой строки ОЗУ.

Коммутатор адреса ОЗУ (КА) состоит из двух частей. Первая часть управляется сигналами РРП и предназначена для передачи в матрицу ОЗУ адреса ячейки, в которой происходит регенерация, а в момент записи-считывания данных — последовательно адреса строки и адреса столбца ячейки ОЗУ; вторая часть — для последовательной передачи в первый адрес строки и столбца ячейки ОЗУ и формирования сигнала ФСТБ, по которому УУ вырабатывает сигнал СТБ.

Для использования в ОС ЭД или отдельных его компонентов написаны два драйвера: записи-считывания ПЗУ и ОЗУ и записи-считывания для ПЗУ. Первый драйвер предназначен для системного устройства, единственное ограничение — файл SWAP должен быть размещен в начале диска, модернизации ОС или ее компонентов при этом не требуется.

Драйвер записи-считывания для ПЗУ может использоваться по алгоритму [7] для программирования отдельных микросхем ПЗУ типа К573РФ4, К573РФ41, К573РФ42, напряжение питания микросхем ПЗУ не повышается.

Принципиальная схема ОЗУ выполнена на 43 микросхемах средней степени интеграции.

Примеры применения ЭД. Основанного на ПЗУ:

инструментальный комплекс для написания и отладки ПО состоит из ЭД (желательно применение одного из ЭД [2, 6]), в котором записана ОС, утилиты для работы с файлами, языковые процессоры, и магнитного носителя, предназначенного для хранения разрабатываемых программ; управляющая ЭВМ. В ЭД ПЗУ записывается монитор ОС, необходимые драйверы внешних устройств и пользовательская управляющая программа, которая может иметь оверлейную структуру (допускаются программы пользователя на языке БЕЙСИК, при

этом в ЭД ПЗУ должен быть записан соответствующий интерпретатор).

630087, Новосибирск, пр. К. Маркса, 20, Новосибирский электротехнический институт; тел. 46-05-53

ЛИТЕРАТУРА

1. Груднин М. М., Сеиченкова А. Ю. Операционная система в ПЗУ // Микропроцессорные средства и системы. — 1986. — № 6. — С. 22—23.
2. Лукьянов Д. А. «Электроника 256К» — эмулятор диска для комплексов на основе микроЭВМ «Электроника 60» и ДВК // Микропроцессорные средства и системы. — 1986. — № 12. — С. 62—64.
3. Сорокин Ю. Ю., Лаврентьев В. В., Максимак С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60М» // Микропроцессорные средства и системы. — 1986. — № 15. — С. 92.
4. Бахмацкий В. Д., Белый В. Г., Раппопорт В. Л., Раппопорт Л. А. Периферийный блок памяти для реализации версии ОС РАФОС // Микропроцессорные средства и системы. — 1986. — № 5. — Вкладка.
5. Зеленко Г. В., Панов В. В., Попов С. Н. Электронный квазидиск для персональной ЭВМ // Микропроцессорные средства и системы. — 1984. — № 4. — С. 79—81.
6. Злотник Е. М., Стежко И. К., Анисенко В. В., Киркоров С. И. «Электронный диск» для вычислительных комплексов на базе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 90—92.
7. Щербаков О. А. Физические основы записи в ПЗУ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 72—75.

Статья поступила 12.08.87

УДК 681.32

К. В. Бехтерев, А. И. Воробьев, А. П. Майоров

ЭМУЛЯТОР ДИСКА В КРЕЙТЕ КАМАК

При проведении научно-технических исследований широко используются комплексы, построенные на основе микроЭВМ «Электроника 60» и аппаратуры, выполненной в стандарте КАМАК. Они имеют в своем составе накопители на гибких магнитных дисках и эксплуатируются с ОС РАФОС. Однако производительность систем, использующих гибкие диски для накопления информации в ходе эксперимента, в значительной степени снижается из-за малой скорости обмена информацией и недостаточной надежности дисков [1, 2].

Устранить указанные недостатки накопителей на гибких магнитных дисках позволяет эмулятор диска, хра-

нящий информацию в дополнительном оперативном ЗУ.

Наиболее просто система обеспечивается дополнительной быстросействующей и надежной памятью, состоящей из четырех модулей КАМАК ОЗУ 16К×24 [3], установленных в четыре последовательные ступени крейта КАМАК. Каждый модуль имеет свои регистры адреса и данных, адреса которых зависят от места установки модуля в крейте. Модули имеют схемы регенерации, обеспечивающие возможность работы с ними как со статической памятью, и допускают автономный режим работы.

```

; ----- ЗАГОЛОВОК -----
.ENABL LC,GBL
; АССЕМБЛЕР MACRO-11 / РАФОС
; *****
; * ДРАЙВЕР ЭМУЛЯТОРА ДИСКА *
; * В КАЧЕСТВЕ ВНЕШНЕЙ ПАМЯТИ ИСПОЛЬЗОВАНЫ *
; * ЧЕТЫРЕ МОДУЛЯ КАМАК ОЗУ 16К*24 *
; *****

```

```

; ----- ОПРЕДЕЛЕНИЯ -----

```

```

.MACRO ERRST ; ПРОВЕРКА ОТВЕТА "X" МОДУЛЯ
BIT $40000,0 CSR ; ОШИБКА "X"?
BEQ ERR ; ДА. ВОЗВРАТ В МОНИТОР
.ENDM ERRST

CSR= 164000 ; АДРЕС РЕГИСТРА СОСТОЯНИЯ И
; УПРАВЛЕНИЯ КОНТРОЛЛЕРА КРЕЙТА
NKX= 1 ; МЕСТО УСТАНОВКИ ПЕРВОГО
; МОДУЛЯ ОЗУ В КРЕЙТЕ
ADR= NKX*40+2+CSR ; АДРЕС РЕГИСТРА АДРЕСА
; ПЕРВОГО МОДУЛЯ

```

```

; ----- ПРОГРАММА -----

```

```

.MCALL .DRDEF
.DRDEF KX,376,(FILST|KX*COD),256.,CSR,0
.DRBEQ KX

MOV KXCQE,R5
MOV BR5,R1 ; ВЗЯТЬ НОМЕР БЛОКА
MOV BR1,R4

.REPT 8.
ASL R1 ; ПОЛУЧИТЬ АДРЕС БЛОКА
; В МОДУЛЕ
.ENDR

BIC $^C300,R4
ASL R4 ; ПОЛУЧИТЬ АДРЕС РЕГИСТРА
ADD $ADR,R4 ; АДРЕСА В МОДУЛЕ
MOV Q#BUFF(R5),R3 ; ПОЛУЧИТЬ АДРЕС БУФЕРА
; ПОЛЬЗОВАТЕЛЯ

MOV $16.,0 CSR ; ЗАГРУЗКА КОДА ФУНКЦИИ
; КАМАК
MOV R1,(R4)+ ; ЗАГРУЗКА АДРЕСА БЛОКА
ERRST

MOV Q#WCNT(R5),R0 ; ПОЛУЧИТЬ СЧЕТЧИК СЛОВ
BEQ DONE
BPL READ
WRITE: NEG R0 ; ПЕРЕПИСАТЬ БЛОК ИЗ
MOV R0,R1 ; БУФЕРА ПОЛЬЗОВАТЕЛЯ
BIC $^C377,R1
10x: MOV (R3)+,BR4
ERRST
DEC R0
BNE 10x

20x: TSTB R1 ; ОЧИСТКА КОНЦА БЛОКА,
BEQ DONE ; ЕСЛИ ОН НЕ ЗАПОЛНЕН
MOV R0,BR4
ERRST
INC R1
BR 20x

READ: MOV BR4,(R3)+ ; ПРОЧИТАТЬ БЛОК В
ERRST ; БУФЕРЕ ПОЛЬЗОВАТЕЛЯ
DEC R0
BNE READ

KXINT:
DONE: CLR Q#WCNT(R5)
BR DONE1
ERR: BIS #NDERRK,0-(R5) ; УСТАНОВИТЬ БИТ ОШИБКИ
DONE1: .DRFIN KX ; ВОЗВРАТ УПРАВЛЕНИЯ
.DREND KX ; В МОНИТОР
.END

```

Обмен информацией между дополнительной памятью и ЭВМ организован стандартными блоками по 256 слов. Память представлена в ОС как устройство с именем КХ емкостью 256 блоков, имеющее каталог файлов. Все операции с КХ аналогичны операциям с другими устройствами прямого доступа ОС.

Устройство КХ поддерживается драйвером (см. рисунок). Идентификатор НКХ определяет номер станции, в которую установлен первый модуль. В крейте КАМАК использован контроллер КК (идентификатор CSR определяет базовый адрес контроллера).

Текст драйвера необходимо оттранслировать совместно с файлом условий системной генерации, скомпоновать и включить в ОС. Для версий ОС, начиная с четвертой и старше, драйвер устанавливается автоматически при загрузке системы. Для более ранних версий драйвер необходимо инициализировать командой IRSTALL. Перед началом работы с дополнительной памятью следует подать команду INIT, по которой инициализируется каталог. После этого устройство КХ готово к работе.

Применение описанного устройства позволило заметно увеличить вычислительную мощность системы, использующей аппаратуру КАМАК. Эксплуатация эмулятора осуществляется на информационно-управляющем комплексе, предназначенном для автоматизации процесса геологоразведочного бурения. Специализированная ОС копируется перед началом работы в устройство КХ. В процессе работы используемые программы и данные считываются и записываются в КХ. Результаты копируются на гибкие диски. Подобная методика использования внешнего ОЗУ позволяет ускорить обмен в 10-15 раз.

198005, Ленинград, 1-я Красноармейская, 1. Ленинградский механический институт; тел. 259 11-54

ЛИТЕРАТУРА

1. Злотник Е. М., Стежко И. К., Аннищенко В. В., Киркоров С. И. «Электронный диск» для вычислительных комплексов на базе микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1986.— № 5.— С. 90—92.
2. Лукьянов Д. А. «Электроника 256К» — эмулятор диска для комплексов на основе микроЭВМ «Электроника 60» и ДВК // Микропроцессорные средства и системы.— 1986.— № 2.— С. 62—65.
3. Устройство запоминающее оперативное 16К×24. Техническое описание и инструкция по эксплуатации 1Г3.065.021 ТО / НТО АН СССР, 1982.— С. 23.

Статья поступила 15.06.87

УДК 681.327.11

В. А. Каннеров, В. В. Роднонов, В. Б. Стригин, Н. Н. Халько

УСТРОЙСТВО СОПРЯЖЕНИЯ ДИСКОВОДА ЕС 5074 С КОНТРОЛЛЕРОМ ДИСКОВОДА PL×45D

МикроЭВМ «Электроника 60» чаще всего используется в системах вместе с НГМД PL×45D и ГМД-7012, подключаемыми через стандартные контроллеры. Однако нередко бывает необходимо применить в системе дисководы других типов, не имеющие стандартных контроллеров и отличающиеся интерфейсными сигналами, например ЕС 5074. С этой целью разработано простое устройство сопряжения, позволяющее подсоединить два дисковода ЕС 5074 к стандартному контроллеру дисковода PL×45D.

Устройство осуществляет взаимное преобразование интерфейсных сигналов дисковода и контроллера. Необходимость в нем обусловлена тем, что дисковод PL×45D с аппаратной точки зрения представляет собой одно внешнее устройство с двумя кассетами для ГМД и двоясным механизмом позиционирования головок, в то время, как ЕС 5074 имеет одну кассету. Это при-

водит к необходимости коммутации интерфейсных сигналов между двумя устройствами ЕС 5074 и одновременного позиционирования их головок. Кроме того, предельной дорожкой для PLX45D является нерабочая дорожка с номером «-1» (при этом головки выведены за пределы области используемых на диске дорожек), а для ЕС 5074 — нулевая рабочая дорожка. Это приводит к неправильной работе механизмов позиционирования при простом подключении сигналов управления головкой без преобразования.

Дисковод PLX45D обрабатывает считанную с диска информацию, выделяя из нее сигналы данных и синхронизации, а НГМД ЕС 5074 таких преобразований не выполняет и, кроме того, не формирует сигналов готовности дисководов из сигналов фотодатчика индекса. И, наконец, дисководы имеют различные временные соотношения между сигналами одинакового назначения.

Устройство сопряжения состоит из четырех функциональных блоков (рис. 1).

Блок состояния дисководов формирует для контроллера сигналы готовности правого (ГПД) и левого (ЛГД) дисков из сигналов фотодатчиков индексов, поступающих из НГМД.

Блок управления позиционированием преобразует сигналы управления головкой, поступающие из контроллера, в сигналы, необходимые для управления ЕС 5074.

Блок коммутации осуществляет связь интерфейсных сигналов в зависимости от состояния шины выбора диска (ВБД).

Блок выделения сигналов данных и синхронизации необходим для первичной обработки считанных с диска данных.

Принципиальная схема устройства сопряжения изображена на рис. 2. Шинные формирователи DD6, DD7 и мультиплексор DD14 коммутируют интерфейсные сигналы

налы в зависимости от состояния шины ВБД с одним из двух дисководов. Сигналы, управляющие позиционированием головки, подаются на оба дисководов одновременно, причем вентили DD13.1...DD13.4 преобразуют контроллерные сигналы «шаг вперед» (ШВП) и «шаг назад» (ШНЗ) в сигналы «шаг» (ШГ) и «направление движения» (НПР). Триггер DD8.1 предназначен для формирования сигнала «предельная дорожка» (ПДО), поступающего в контроллер. При установке головки дисковод ЕС 5074 на нулевую дорожку на выходах дисководов ДО 0 и ДО 1 появится низкий уровень, индицирующий это состояние и запрещающий прохождение очередного ШНЗ, соответствующего перемещению головок на предельную нерабочую дорожку PLX45D, на механизм позиционирования.

При появлении в этом состоянии очередного сигнала ШНЗ триггер DD8.1 взведется и передает в контроллер сигнал ПДО. Первый после этого ШВП также не будет передан в механизм позиционирования, а вызовет сброс триггера DD8.1 и, соответственно, разрешение прохождения в НГМД последующих ШВП.

Формирование сигналов готовности дисков, необходимых для работы контроллера, осуществляется следующим образом. Импульсы фотодатчиков индекса дисков запускают одновибраторы DD1.1 и DD1.2 и, если скорость вращения дисков меньше номинальной, определяемой длительностью выходных импульсов одновибраторов, триггеры DD2, DD3 не взводятся. По достижении дисками номинальной скорости вращения триггеры DD2, DD3 последовательно взводятся. При этом на выходах вентилей DD4.1 и DD4.2 появляются сигналы готовности ЛГД и ГПД дисков после 2,5 оборотов диска с номинальной скоростью.

Запись информации на диск происходит методом двойной частоты, т. е. любое изменение направления намагниченности диска представляет собой бит Лог.1, причем бит данных записывается после бита синхронизации.

В соответствии с этим сигнал «данные воспроизведения» (ДВС) на выходе НГМД ЕС 5074 представляет собой смесь сигналов синхронизации и данных, причем каждому изменению намагниченности диска, т. е. фронту или срезу сигнала (каждому биту Лог.1, считанному с диска), поступающему с головки, соответствует уже сформированный импульс длительностью 400 нс.

Блок первичной обработки данных выделяет из сигнала ДВС биты информации и биты синхронизации «выделенные данные чтения» (ВДЧ) и «выделенная считанная синхронизация» (ВСС) соответственно, а также стробирующий их сигнал «синхронизация счета» (ССЧ).

Каждый импульс данных воспроизведения с выбранного диска запускает одновибраторы DD15.1 и DD15.2. Выходной сигнал одновибратора DD15.2 с длительностью 800 нс сравнивается на вентиле DD16.2 с выходным сигналом синхрогенератора, построенного на триггере Шмитта по схеме генератора, управляемого напряжением. При наличии между ними фазового сдвига сигнал на выходе DD16.2 после интегрирования осуществляет подстройку частоты генератора до устранения сдвига фаз.

Разделение битов синхронизации и данных осуществляется следующим образом. Выходной импульс одновибратора DD15.1 длительностью 400 нс взводит один из триггеров DD8.2, DD9.2 в зависимости от состояния триггера DD10.2. Если последний взведен, то устанавливается DD8.2, что соответствует единичному биту синхронизации, а если DD10.2 сброшен, устанавливается DD9.2, что соответствует единичному биту данных. Содержимое триггеров DD8.2 и DD9.2 в каждом такте переписывается в триггеры DD9.1, DD10.1 и через буферные инверторы DD4.3 и DD4.4 передается в контроллер вместе со стробирующим сигналом ССЧ, представляющим сигнал на выходе триггера DD10.2, задержанный на элементах DD12.2, DD12.3 на 1,5 мкс. Фази-

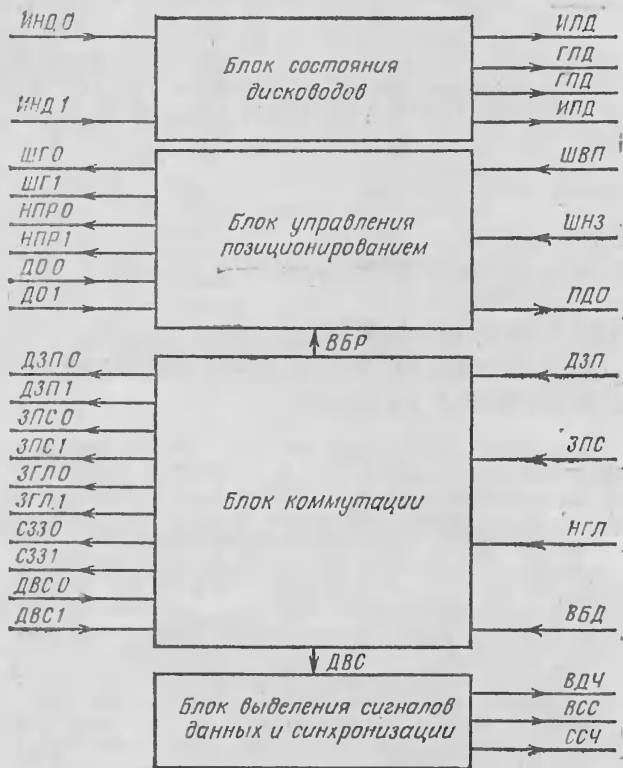


Рис. 1. Структурная схема устройства сопряжения

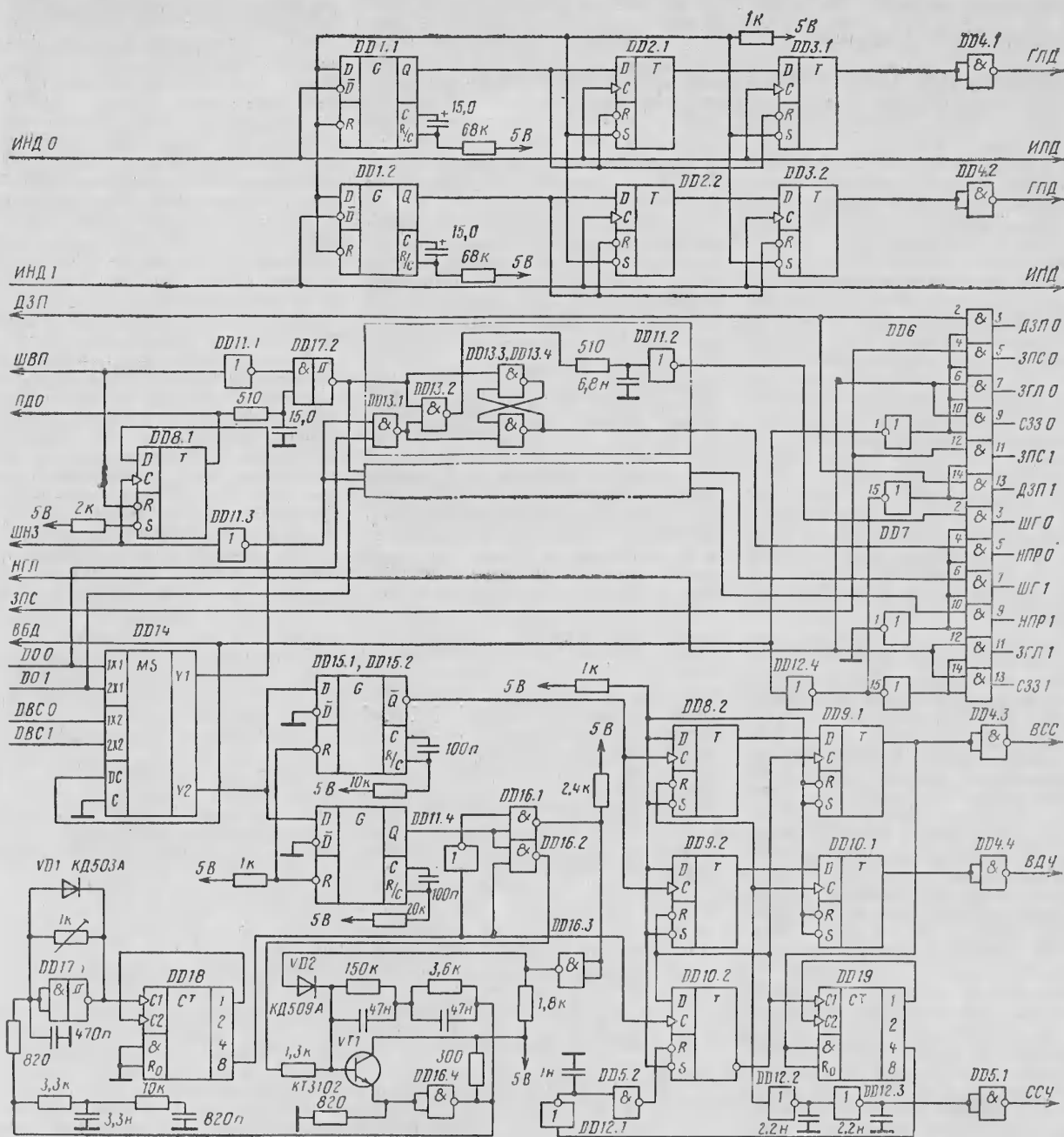


Рис. 2. Принципиальная схема устройства сопряжения:

DD1, DD15 — К155АГ3; DD2, DD3, DD8, DD9, DD10 — К155ТМ2; DD4, DD5 — К155ЛА12; DD6, DD7 — К155ЛП11; DD12 — К155ЛН1; DD13 — К155ЛЛ3; DD14 — К531КП11; DD16 — К155ЛЛ8; DD17 — К155ТЛ3; DD18, DD19 — К155МЕ5

ровка триггера DD10.2, соответствующая правильному разделению битов данных и синхронизации, осуществляется с выхода триггера DD9.1 через микросхемы DD19, DD12.1, DD5.2.

Для повышения надежности работы блока сопряжения предусмотрены согласующие нагрузочные элементы для входных интерфейсных сигналов.

При использовании описанного модуля сопряжения для подключения к стандартному контроллеру диско-

вода PLX45D двух НГМД ЕС 5074 не требуется вносить изменения в контроллер и программное обеспечение. И хотя паспортная частота позиционирования головки дисководов ЕС 5074 в два раза ниже, чем у PLX45D, практика их длительной эксплуатации показала допустимость работы при повышенной частоте без сбоев.

Телефон 324-81-54, Москва

Сообщение поступило 19.04.87

СИМВОЛЬНО-ГРАФИЧЕСКИЙ ДИСПЛЕЙ НА БАЗЕ ГАЗОРАЗРЯДНОЙ ИНДИКАТОРНОЙ ПАНЕЛИ

В микропроцессорных устройствах в качестве средств отображения информации все более широкое применение находят матричные индикаторные панели, благодаря ряду преимуществ плоских индикаторов над средствами отображения информации на основе электронно-лучевых трубок. К ним относятся: меньшие габаритные размеры и масса при больших размерах поля отображения; плоскость конструкции экрана; отсутствие геометрических и нелинейных искажений изображения; высокая надежность; невысокие питающие напряжения; высокая эффективность преобразования электрической энергии в световую; удобство адресации информации в ячейки индикатора. Все эти преимущества играют важную роль, особенно при создании средств отображения информации для портативных и малогабаритных устройств.

Промышленностью выпускаются несколько газоразрядных индикаторных панелей постоянного тока (информационная емкость 10 000 элементов). Панель ГИП-10 000 — это матрица светонезлучающих ячеек, образованных взаимно-перпендикулярными пересекающимися вертикальными и горизонтальными электродами, пространство между которыми заполнено смесью инертных газов. При подаче на ячейку напряжения постоянного тока в газе между двумя электродами возникает тлеющий разряд, сопровождающийся свечением; появляется изображение на экране индикатора. Однако устройство панели не позволяет подавать напряжение одновременно на все ячейки, требующие засветки. Для получения изображения на панелях постоянного тока с внешней адресацией схема развертки последовательно подает группы горизонтальных или вертикальных электродов к одному из полюсов источника напряжения постоянного тока. Еще одна группа электродов подключается к схеме управления синхронно с разверткой. Рассматриваемая панель постоянного тока не имеет внутренней памяти, поэтому для хранения отображаемой информации схема управления должна содержать блок ОЗУ.

Индикаторные газоразрядные модули ИМГ-1(02) с дополнительными схемами управления предназначены для отображения информации в виде знаков, символов, графиков и гистограмм [1].

Состав модуля (см. рисунок): газоразрядная индикаторная панель ГИП-10 000;

блоки анодных и катодных ключей (АК и КК), предназначенные для управления ячейками;

дешифратор (КД) для селективного последовательного управления катодными ключами (при подаче на его вход линейного изменяющегося 8-разрядного двоично-десятичного кода позволяет получить растровый режим работы индикатора);

устройство защиты (УЗ), предотвращающее повреждение ячеек чрезмерным током в случае прекращения смены кодовых комбинаций на входах X.

Индикатор ИМГ-1-03 выпускается промышленностью на основе ИМГ-1-02. Он обеспечивает последовательный прием адреса, знакоместа и запись знака по этому адресу и позволяет последовательно увеличивать или уменьшать на единицу ад-

рес знакоместа по вертикали и горизонтали.

Разработанный на базе газоразрядного индикаторного модуля ИМГ-1-01(02) дисплей предназначен для отображения буквенно-цифровой и произвольной графической информации в различных микропроцессорах или ОЭВМ.

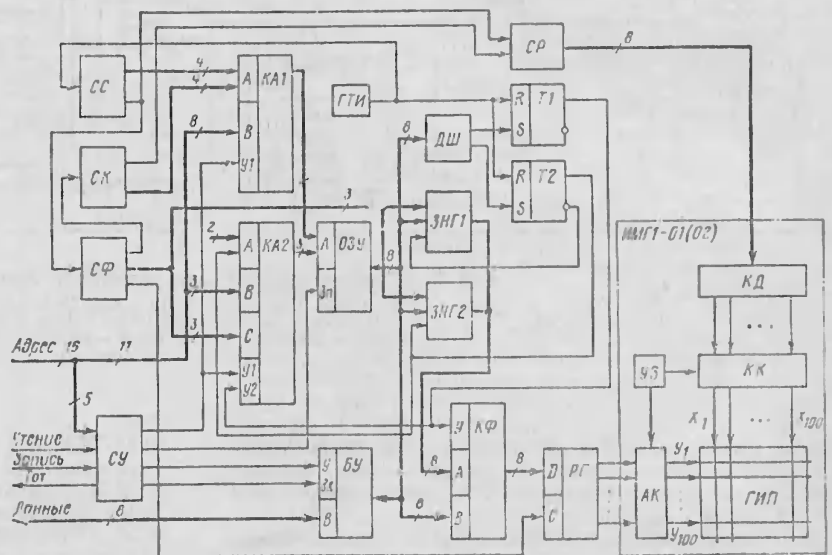
В разработанном дисплее применяется растровый способ формирования изображения с вертикальной разверткой (по столбцам). Катодами (вертикальными электродами) панели управляет схема развертки, входящая в состав ИМГ-1-02, а аноды панели (горизонтальные электроды) подключены к схеме управления.

Дисплей позволяет отображать буквенно-цифровую, графическую информацию, а также их комбинацию. Поле отображения дисплея (96×96 точек) разбито на 192 знакоместа (12 строк по 16 символов), каждое из которых воспроизводит либо символ, либо элемент графической информации. Размер знакоместа (графического элемента) 6×8 точек.

При формировании символов используется матрица 5×7 точек. Непользуемые точки знакоместа соз-

Технические характеристики

	ИМГ-1-03	Разработанный дисплей
Число символьных строк	10	12
Число знакомест в строке	16	16
Формат знакоместа, точек	6×10	6×8
Число символов в наборе	96	254+256
Поле графической информации, точек	Нет	96×96
Объем буферного ОЗУ, Кбайт	1	2
Магистраль обмена	ИРПР	И-41
Потребляемая мощность от источников питания, Вт (В)	18(+12) 10(+5)	8(+245) 2,5(+5, ТТЛ), 0,5(+5, КМОП) 145×145×70 1,0
Габаритные размеры, мм	145×145×100	
Масса, кг	1,4	



Структурная схема модуля

дают промежутки. Столбец, состоящий из вертикальных символов, принадлежащих одной колонке, формируется схемой управления и заносится в 96-разрядный регистр, выход которого управляет анодными ключами индикатора. Информация регистра синхронно изменяется с каждым шагом развертки и последовательно отображает вертикальные фрагменты всех символов.

В графическом режиме в знакоместах вместо символов — элементы графической информации. Вертикальные фрагменты при отображении символической информации формируются с помощью знакогенератора, а графическая информация извлекается прямо из блока памяти дисплея.

Графический элемент одного знакоместа состоит из шести фрагментов (столбцы по восемь точек). Информация о фрагментах каждого графического элемента хранится в шести графических страницах буферного ОЗУ в байтах.

Каждой точке графического фрагмента соответствует бит 8-разрядного информационного слова ячейки памяти графической страницы (метод битовой карты).

Рассмотрим работу устройства.

При совпадении адресов и устройства схема управления (СУ) посылает сигнал У1, переключающий адресные коммутаторы (КА1, КА2), которые выбирают в качестве источника адреса для буферного ОЗУ адресную шину управляющей микроЭВМ. В зависимости от операции (чтение или запись данных) схема управления выработывает СУ двунаправленными буферными усилителями (БУ), обеспечивающими необходимое направление передачи информации. Если в момент обращения микроЭВМ к дисплею происходит передача кодов из блока памяти на индикацию, то для исключения появления на экране хаотических помех из СУ поступает сигнал Гот, останавливающий микроЭВМ до завершения этого процесса.

Пересчетная схема, состоящая из счетчиков символьных строк (СС), колонок (СК) и фрагментов (СФ), формирует адреса буферного ОЗУ в процессе отображения информации. Выходы СС и СК образуют восемь младших линий адреса, поступающего через КА1 на адресный вход блока памяти. Три старшие адресные линии блока памяти идут с КА2. В зависимости от режима работы КА2 выбирает в качестве источника либо константу со входа А (символьный режим), либо выходы СФ со входа С (графический режим).

На знакогенератор ЗНГ1 подаются код символа с выхода блока памяти и код фрагмента с СФ. Фрагмент символа через коммутатор фрагментов (КФ) поступает на вход сдвигового регистра, образованного 12 8-разрядными ступенями. В ре-

гистре хранится информация об одном отображаемом столбце панели, а 96 его выходов управляют соответствующими анодными ключами индикатора. После занесения в регистр 12 фрагментов очередного отображаемого столбца следует пауза, обеспечивающая надежное зажигание ячеек панели и хорошую яркость их свечения.

Для отображения графического элемента на экране необходимо в соответствующую ячейку памяти основной символической страницы поместить код OFFH [2]. При его передаче на индикацию сигнал на первом выходе ДШ установит триггер Т1 в состояние Лог.1 и переключит КА2 и КФ, на три старшие адресные линии блока памяти с СФ поступит код соответствующей графической страницы. Графический фрагмент через КФ попадает на вход сдвигового регистра (СФ). При заполнении всей основной символической страницы кодом OFFH получается чисто графический режим работы индикатора. Используя его, можно программными методами создавать различные изображения (произвольно расположенные символы разнообразных форм и размеров, пиктограммы, графики). Ограничения накладывает только разрешающая способность экрана (96×96 точек). Режим работы дисплея можно изменить в любом месте экрана, он позволяет совмещать графическую информацию с символической (в соответствующие ячейки символической страницы вместо кода OFFH помещаются коды требуемых символов). Такой режим упрощает процесс передачи символической информации и экономит процессорное время (символы формируются с помощью знакогенератора, а не программно).

Буферное ОЗУ дисплея имеет 8-разрядную организацию, поэтому одно информационное слово адресует не

более 256 различных символов знакогенератора. Для расширения набора отображаемых символов служит знакогенератор ЗНГ2 и триггер Т2.

При передаче кода OFFH на отображение на втором выходе ДШ появляется сигнал, перебивающий триггер Т2. Вследствие изменения состояния триггера Т2 изменяется младший разряд 3-разрядной константы, подаваемой на старшие адресные входы блока памяти через коммутатор КА2. Это приводит к переключению символической страницы, и для отображения поступает код из ячейки памяти, расположенной во второй символической странице. Такой способ расширения набора отображаемых символов может быть использован и в дисплеях на ЭЛТ.

Разработаны два варианта дисплея: на ТТЛ микросхемах серии К533 и КМОП микросхемах серии К561.

Описанный дисплей применен в портативной системе отладки «САД-КО» [3]. Его эксплуатация подтвердила надежность в различных микропроцессорных системах.

290646, Львов-13, ул. Мира, д. 12, ЛПИ, РТФ, тел. 34-12-33

ЛИТЕРАТУРА

1. Вуколов Н. И., Михайлов А. Н. Знакосинтезирующие индикаторы. Справочник. — М.: Радио и связь, 1987. — С. 518—521.
2. А. с. 1322367 (СССР). Устройство для отображения информации / Р. М. Гайдучок, Г. А. Шумский — Оpubл. 1987. Бюл. № 25.
3. Гайдучок Р. М., Шумский Г. А. Портативная система отладки — тестер «САДКО» // Микропроцессорные средства в системах. — 1987. — № 1. — С. 63—65.

Статья поступила 06.10.87

УДК 681.3.01 : 51

А. В. Новиков, В. П. Климов, В. Н. Смирнов, В. К. Ковальков

УСТРОЙСТВО ПОСЛЕДОВАТЕЛЬНОГО ИНТЕРФЕЙСА МУЛЬТИПЛЕКСНОГО КАНАЛА ИНФОРМАЦИОННОГО ОБМЕНА

В распределенных информационно-измерительных и вычислительных системах с централизованным методом управления находят широкое применение мультиплексные каналы информационного обмена (МКИО). В соответствии с ГОСТ 26.765.52-87 [1] МКИО содержит магистральную шину последовательной передачи данных, к которой через отводы можно подключить до 31 терминала. Управление МКИО осуществляется с помощью контроллера. Передача управления соответствующему терминалу происходит пересылкой управляющего маркера.

Рассматриваемое устройство последовательного интерфейса (УПИ) позволяет организовать информационный обмен между несколькими микроЭВМ «Электроника 60М», выполняющими роль терминалов и объединенными в локальную сеть. УПИ работает в режимах контроллера (К) или оконечного устройства (ОУ) МКИО и обеспечивает согласование интерфейсов по ГОСТ 26.765.52-87 и ОСТ 11.305.903-80. Форматы сообщений, реализуемые УПИ, представлены на рис. 1. Протокол информационного обмена реализуется в УПИ в основном аппара-

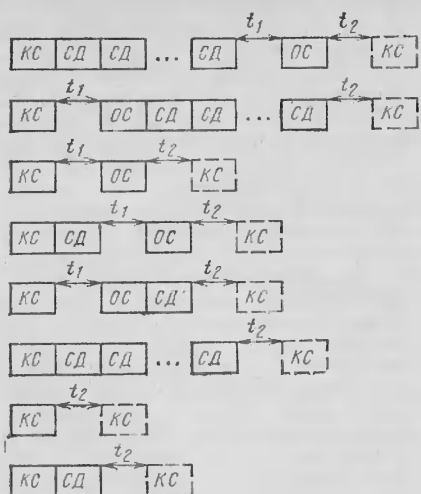


Рис. 1. Основные форматы сообщений, реализуемые УПИ:

КС — командное слово; СД — слово данных; ОС — ответное слово; t_1 — время ожидания контроллером ответного слова (не менее 12 мкс); t_2 — пауза между сообщениями формируемыми контроллером (не менее 4 мкс)

турными средствами, что позволяет максимально разгрузить от них процессор терминала МКИО.

Структура УПИ (рис. 2) насчитывает следующие основные блоки:

адаптер канала (АК) БИС К588ВГ6, предназначенный для прямого и обратного преобразования 16-разрядных слов параллельного формата в последовательный бифазный код и идентификации принимаемой из МКИО информации, определяет тип слова (слово данных, командное слово), тип команды (передача данных, управление), распознает адреса, присвоенные терминалам в МКИО, временно хранит информацию во внутренних регистрах [2, 3];

блок определения достоверности (БОД) командных (ответных) слов и слов данных принимает от АК сигналы, характеризующие передаваемую по МКИО информацию (VA, P, CD, OP, TD, GD, OB), и вырабатывает следующие сигналы: достоверное командное (ответное) слово (ДКОС) — при обнаружении соответствующих слов, предназначенных дан-

ному терминалу; достоверное слово данных (ДСД);

блок микропрограммного управления (БМПУ) управляет блоками УПИ в соответствии с принятыми алгоритмами функционирования;

блок прямого доступа к памяти (БПДП) связывает УПИ с системным каналом терминала МКИО и включает контроллер прямого доступа к памяти (КПДП) БИС К588ВТ2, программатор КПДП (ПРК), регистры старших и младших разрядов адреса (РСРА, РМРА), числа слов данных (РЧСД), слова состояния контроллера (РССК), блок режима контроллера (БРК);

селектор адреса (СА) БИС К588ВТ1 выполняет адресные обращения к программно-доступным регистрам УПИ со стороны процессора терминала МКИО.

Архитектура УПИ представлена рядом программно-доступных регистров. Регистр командных слов (РКС) является доступным процессору терминала по записи и предназначен для записи слов управления, оп-

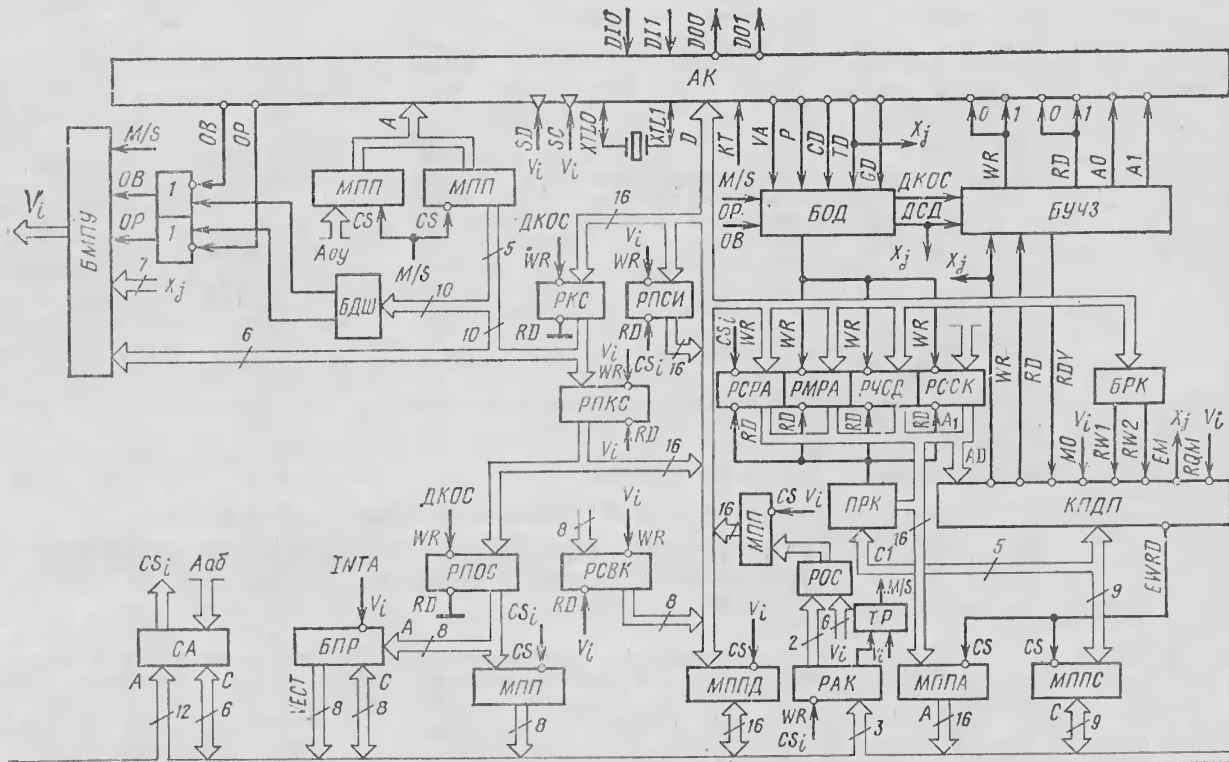


Рис. 2. Структурная схема УПИ:

АК — адаптер канала; БОД — блок определения достоверности; МПП — магистральный приемопередатчик; МППД — магистральный приемопередатчик слова данных; БМПУ — блок микропрограммного управления; БДШ — блок дешифрации; БУЧЗ — блок управления чтением-записью; РКС — регистр командных слов; РПСИ — регистр последнего командного слова; РПСИ — регистр принятой служебной информации; РПОС — регистр принятого ответного слова; РСВК — регистр слова встроенного теста; РСРА, РМРА — регистры старших и младших разрядов адреса; РЧСД — регистр числа слов данных; РССК — регистр слова состояния КПДП; РОС — регистр ответного слова; РАС — регистр абонента канала; МППА — магистральный приемопередатчик адреса КПДП; МППС — магистральный приемопередатчик сигналов управления КПДП; ТР — триггер режима УПИ; ПРК — программатор КПДП; БРК — блок режима КПДП; СА — селектор адреса; БПР — блок прерываний; КПДП — контроллер прямого доступа к памяти

ределяющих начальный адрес микропрограмм, и командных слов, предназначенных для передачи в МКИО (в режиме оконечного устройства в данный регистр заносятся командные слова, принимаемые из МКИО).

Регистр принятого ответного слова (РПОС), доступный процессору по чтению, служат для записи принимаемых ответных слов. Регистр принятой служебной информации (РПСИ) доступен процессору по чтению и предназначен для записи ответных слов, слов встроенного контроля, последних командных слов, принимаемых из МКИО.

Рассмотренные регистры используются при работе УПИ в режиме контроллера МКИО. Регистры, входящие в состав блока прямого доступа к памяти, доступны процессору по записи и служат для записи информации, используемой для программирования КППД при работе УПИ в режиме контроллера МКИО. В режиме оконечного устройства в данные регистры заносится информация, содержащаяся в соответствующих полях командного слова. Запись производится по сигналу ДКОС.

Регистр абонента канала (РАК) доступен процессору терминала по записи и предназначен для установки флагов: «абонент занят», «неисправность абонента» при работе УПИ в режиме оконечного устройства. Остальные регистры УПИ являются программно-недоступными и служат для хранения информации, необходимой для функционирования устройства.

В режиме оконечного устройства УПИ по командам контроллера осуществляет двунаправленный информационный обмен по МКИО, обеспечивает обмен информацией с системным каналом терминала МКИО, выполняет ряд команд управления, производит самоконтроль с формированием слова встроенного теста.

Работа в данном режиме начинается при приеме из МКИО командного слова. Адаптер канала декодирует его и, в случае достоверности и принадлежности данному терминалу, вырабатывает сигналы CD, P, VA (OP в случае работы в общем режиме). Блок управления чтением записью (БУЧЗ) по сигналу ДКОС читает командное слово из буфера адаптера канала и записывает его в РКС. Параллельно разряды 10...19 командного слова заносятся в регистры РМРА и РЧСД блока ПДП, а разряд 9 поступает на БРК, который устанавливает сигналы RW1, RW2, определяющие режим работы КППД. Загрузка регистра старших разрядов адреса производится предварительно адресным обращением процессора терминала МКИО. По получении ответов от указанных регистров программатор блока ПДП загружает

внутренние регистры КППД по шинам A1, C1.

С выхода регистра командного слова на БМПУ подаются пять младших разрядов и разряд 9 командного слова. С АК поступают признаки режима МКИО: передача слов данных/режим управления ОВ, общий режим ОР, а с триггера режима (TR) выдается признак режима работы УПИ — контроллер/оконечное устройство (M/S). В БМПУ указанные разряды и признаки дешифруются, определяется начальный адрес и запускается микропрограмма, реализующая тот или иной алгоритм работы УПИ в данном режиме.

В режиме приема информации из МКИО алгоритм работы следующий. Вначале БМПУ тестирует разряды РАК, соответствующие аномальным признакам «абонент занят», «неисправность абонента». При их наличии УПИ ожидает окончания передачи информации, не производя инициализации устройства ПДП, и выдает в МКИО ответное слово с установленными разрядами 17 и 19. В отсутствие указанных признаков УПИ переходит в режим ожидания слов данных. После получения достоверного слова данных, о чем свидетельствует сигнал ДСД блока определения достоверности, проверяется признак окончания программирования КППД и БМПУ выдает запрос на ПДП (RQM), поступающий на КППД и инициализирующий его работу. Получив от КППД сигнал «чтение регистра» (RD), формируемый в ходе выполнения цикла ПДП, блок БУЧЗ читает буфер АК и выдает сигнал «ответ регистра» (RDY), по которому КППД заканчивает цикл ввода слова данных в ОЗУ терминала МКИО. Каждый последующий цикл ПДП начинается после поступления в УПИ достоверного слова данных путем формирования БМПУ сигнала «продолжение ПДП» на входе «режим работы» МО КППД. В ходе приема слова данных происходит декрементирование счетчика числа слов данных КППД, программируемого информацией о числе слов данных, которые должны быть приняты из МКИО и переданы в системный канал терминала. Если поступление информации прекращается до обнуления данного счетчика или она продолжается поступать после его обнуления, то в РОС заносится признак «ошибка в сообщении» (разряд 9), а в регистр слова встроенного контроля (РСВК) — соответственно признаки «Количество слов <», «Количество слов >». В обоих случаях ответное слово в МКИО не выдается. При обнулении указанного счетчика (EM=1) и освобождении МКИО (TD=0) УПИ выдает ответное слово без установленного аномального признака, а БМПУ формирует сигнал «окончание микропрограммы», после чего УПИ переходит в исходное состояние.

В режиме передачи информации с терминала в МКИО, как и в режиме приема, БМПУ тестирует РАК на наличие аномальных признаков. При их наличии УПИ формирует ответное слово с соответствующими признаками и, в случае освобождения МКИО, выдает его без слов данных. Если МКИО не освобождается, то в РОС устанавливается признак «ошибка в сообщении», в РСВК — признак «Количество слов >» и ответное слово в МКИО не выдается. При отсутствии аномальных признаков в РАК и своевременном освобождении МКИО выдается ответное слово, а БМПУ проверяет признак окончания программирования КППД. Его отсутствие расценивается как неисправность УПИ и в РОС устанавливается соответствующий признак. По указанному признаку БМПУ инициализирует работу КППД, выставляя сигнал RQM. Получив от КППД сигнал «запись регистра» (WR), формируемый в ходе выполнения цикла ПДП, БУЧЗ обеспечивает запись буфера АК и выдает сигнал RDY, по которому КППД завершает цикл ввода слова данных из ОЗУ абонента МКИО. Затем производится старт слова данных (SD) и передача его в МКИО. Аналогичным образом выдается весь информационный массив, причем каждый последующий цикл ПДП начинается при приеме АК собственного слова данных в режиме эхоконтроля.

В режиме управления УПИ реализует тот или иной алгоритм, соответствующий принятой от контроллера МКИО команде, и выдает ответное слово. В УПИ предусмотрена автоматическая блокировка передатчика в случае, если время передачи сообщения превышает 800 мкс (число передаваемых слов > 32). При этом в РСВК устанавливается соответствующий признак. Блокировка снимается при поступлении в УПИ достоверного командного слова.

В режиме контроллера УПИ организует двунаправленный информационный обмен с терминалами МКИО, осуществляет прием и анализ ответных слов от оконечных устройств терминалов МКИО с выдачей вектора прерывания центральному вычислительному устройству, к которому подключен контроллер МКИО; инициализирует оконечные устройства МКИО на выполнение команд управления путем передачи соответствующих командных слов.

Особенность функционирования УПИ в данном режиме по сравнению с режимом оконечного устройства заключается в том, что загрузка регистров блока ПДП (РСРА, РМРА, РЧСД) и РКС осуществляется адресным обращением со стороны центрального вычислительного устройства, после чего программатор блока ПДП начинает программирование КППД.

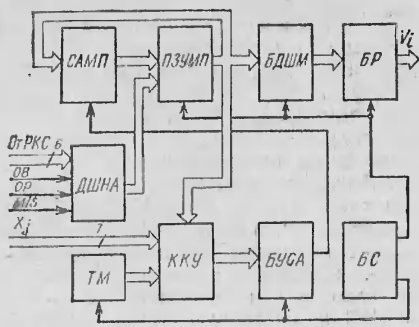


Рис. 3 Структурная схема блока МПУ

На БМПУ поступает пять младших разрядов и разряд 9 командного слова, с выхода блока дешифрации (БДШ) — признаки режима управления (ОВ) и общего режима (ОР), полученные в результате дешифрации 4...8 и 10...14 разрядов командного слова, с ТР — признак режима работы УПИ, соответствующий режиму контроллера (М). В результате дешифрации указанных признаков определяется начальный адрес микропрограммы и БМПУ переходит к формированию требуемого алгоритма работы. В МКМО выдается командное слово и УПИ осуществляет прием-передачу информационной посылки, так же как и в режиме окончательного устройства.

Другая особенность определяется необходимостью контроля времени получения ответных слов от окончательных устройств и их анализа. Если ответное слово отсутствует в течение

12 мкс, то УПИ информирует об этом центральное вычислительное устройство выдачей вектора прерывания «отсутствие ответного слова». Для определения причины его отсутствия контроллер может послать в окончательное устройство командное слово, содержащее команду «передать ответное слово».

При своевременном получении ответного слова УПИ анализирует его с выдачей векторов: «ответное слово без аномальных признаков», «абопепт занят». Блок прерываний (БПР) выполнен на основе БИС К588ВН1, работающей в режиме выдачи вектора с внешней шины. В остальном работа в режиме контроллера аналогична работе УПИ в режиме окончательного устройства.

Управляющим устройством УПИ является БМПУ, структура которого (рис. 3) включает: дешифратор начального адреса микропрограммы (ДШНА), ПЗУ микропрограммы (ПЗУМП), счетчик адреса микропрограммы (САМП), таймер с выдержками 2, 4, 12, 20 мкс (ТМ), коммутатор кода условия (ККУ), блок управления следующим адресом (БУСА), блок дешифрации микрокоманд (БДШМ), блок выходных регистров (БР) и блок синхронизации (БС), синхронизирующий работу остальных блоков. Рассматриваемое устройство может работать в двух режимах — последовательной выборки микрокоманд из ПЗУМП и перехода по адресу. В первом режиме счетчик адреса осуществляет последовательный перебор адресов микропрограммы. При этом из ПЗУМП выбираются определенные микро-

команды, в результате дешифрации которых вырабатываются управляющие сигналы. Во втором режиме на входах счетчика адреса микропрограммы устанавливается адрес из ПЗУМП, по которому должен быть осуществлен переход при невыполнении условия Xj. Блок управления следующим адресом в этом случае ведет асинхронную запись указанного адреса в счетчик адреса микропрограммы. При выполнении условия Xj адрес не записывается, а вырабатывается сигнал на инкрементирование указанного счетчика. Условия перехода подразделяются на два вида — с временной выдержкой и без нее.

УПИ реализовано на микросхемах серий К588, К564, К556, К1533 и размещено на трех двусторонних печатных платах размерами 140×150 мм. Телефон 158-44-59, Москва

ЛИТЕРАТУРА

- ГОСТ 26.765.52-87. Интерфейс магистральный последовательный систем электронных модулей. — М.: Стандартиздат, 1988.
- Хвош С. Т., Варлинский Н. Н., Попов Е. А. Микропроцессоры и микроЭВМ в системах автоматического управления. Справочник. — Ленинград: Машиностроение, 1987. — 639 с.
- Хвош С. Т., Смолев В. Б., Сухопаров А. И. Комплект БИС для организации мультиплексных каналов межмодульного обмена информацией // Микропроцессорные средства и системы. — 1984. — № 3, С. 18—23.

Статья поступила 18.06.87

УДК 681.323

С. Ю. Прозоров, В. В. Яковлев

СОПРЯЖЕНИЕ УСТРОЙСТВА СМ5211 С ИНТЕРФЕЙСОМ МИКРОЭВМ «ЭЛЕКТРОНИКА 60».

Устройство внешней памяти на кассетной магнитной ленте (УВПК) СМ5211 предназначено для работы в составе управляющих вычислительных комплексов (УВК) СМ ЭВМ. В состав устройства входят два кассетных накопителя на магнитной ленте СМ5206.

Для использования УВПК СМ5211 в составе УВК на базе микроЭВМ «Электроника 60» разработано устройство сопряжения (далее — согласователь).

Габаритные размеры согласователя не более: длина 252 мм, ширина 143 мм, высота 12 мм. Питание от источника постоянного тока напряжением $5 \text{ В} \pm 5\%$. Максимальный потребляемый электрический ток не более 2 А. При работе в составе микроЭВМ согласователь управляется по программному каналу с прерываниями. Вектор прерывания 260.

В состав согласователя входят: регистр команд и состояния; буферный регистр данных; постоянное запо-

минающее устройство; логические схемы управления и адресации канала ЭВМ; логические схемы управления кассетными накопителями; логическая схема прерывания.

Согласователь имеет два адресуемых регистра: регистр команд и состояния РКС — 177500; буферный регистр данных БРД — 177502; ПЗУ с программой-загрузчиком емкостью 256 слов, позволяющей осуществлять загрузку с кассетной магнитной ленты, перфоленты, гибкого диска. Адреса постоянной памяти 174000...174776. Разработана техническая документация.

Для работы используются система программирования QUASIC* и тест-мониторная система СТС, разработанная в Карельском филиале АН СССР. В комплекте УВПК СМ5211 имеется тест проверки на ЭВМ СМ4, адаптированный для микроЭВМ «Электроника 60» и включенный в состав системы СТС. Тесты, поставляемые с микроЭВМ, также включены в систему СТС.

185610, Петрозаводск, ул. Пушкинская, 11, Карельский филиал АН СССР, отдел математических методов: тел. 6-63-04

Сообщение поступило 28.12.87

* Подольский Л. И., Ляковский А. П. — Система программирования QUASIC-2 для микроЭВМ // Микропроцессорные средства и системы. — 1987. — № 2. — С. 9—11.

В. Г. Васильев, А. О. Воробьев, Н. Т. Задорожная

ОДНОПЛАТНЫЙ ЭМУЛЯТОР ДИСКА

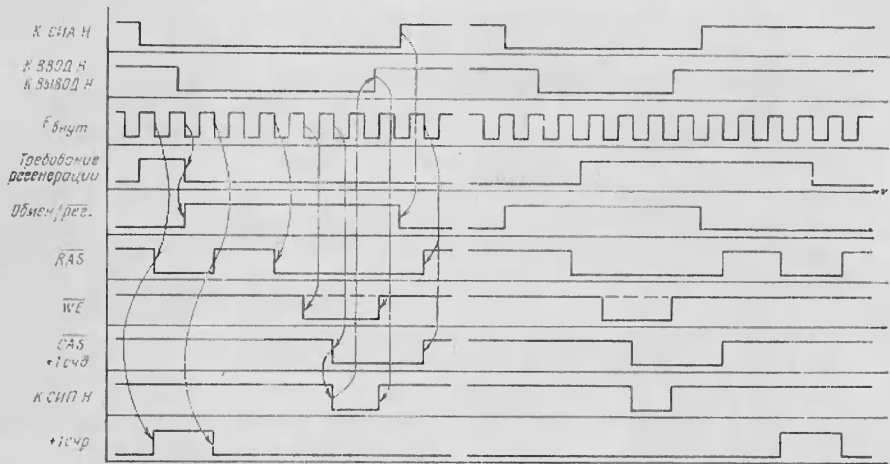
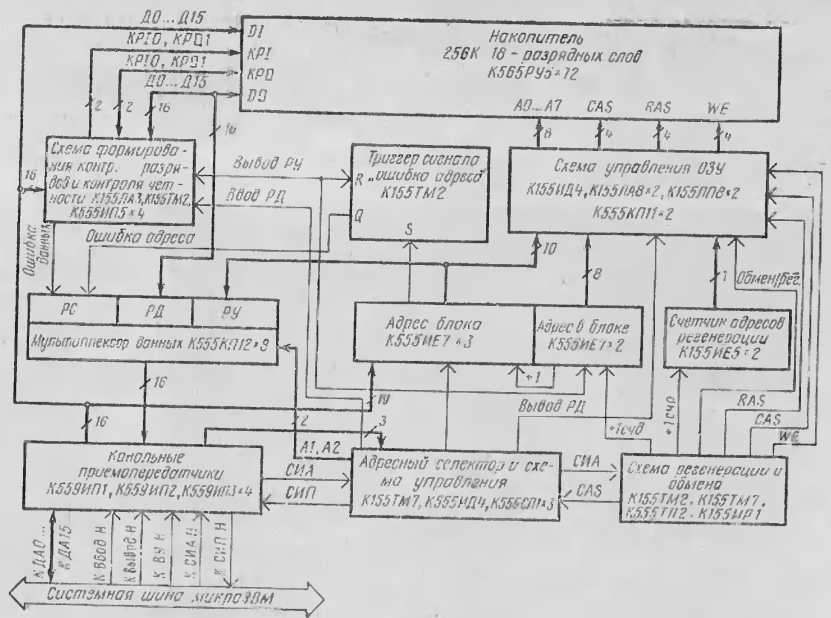
В работах [1—3] для применения в вычислительных комплексах на основе микроЭВМ ряда «Электроника 60» и ДВК рассмотрены электронные эмуляторы гибкого магнитного диска (квазидиска), имеющие существенный недостаток: для их установки требуется дополнительный субблок или самостоятельный конструктив, так как они выполнены из двух и более модулей. Это не позволяет применять их в тех изделиях, где требуется значительная смонтер квазидиска и габаритные размеры играют первостепенную роль.

В настоящей работе рассмотрен одноплатный квазидиск (рис. 1) емкостью 256 К 16-разрядных слов, устанавливаемый непосредственно в субблок микроЭВМ (соединительные кабели исключаются, что повышает надежность и не ограничивает максимальной скорости действия шины микроЭВМ).

В качестве ИС ОЗУ применены ЗУ К565РУ5, регенерация ОЗУ автономная. В одном цикле регенерируется одна строка ОЗУ. Циклы регенерации повторяются через 15 (7) мкс. Таким образом, полный цикл регенерации выполняется за 2 (1) мс (в скобках указаны времена для микросхем К565РУ5Д). Конфликтные ситуации между циклами регенерации и обращения к ОЗУ со стороны микроЭВМ разрешает схема управления регенерацией и обменом.

При временном наложении циклов регенерации и обмена цикл, начавшийся первым, выполняется до конца, после чего выполняется второй. Если требование на регенерацию и обмен возникают одновременно, предпочтение отдается регенерации, т. е. регенерация имеет приоритет перед обращением к ОЗУ со стороны микроЭВМ (рис. 2).

Для управления квазидиском и обменом информацией с микроЭВМ служат три регистра (последовательно расположенных в строке ввода-вывода микроЭВМ): состояния (РС),



```

1) ФОРМАТИЗАЦИЯ КВАЗИДИСКА
.....
EDCSR = 177720 ; АДРЕС РС КВАЗИДИСКА
MAXNBL = 176000 ; НАЧАЛЬНОЕ ЗНАЧЕНИЕ РУ ДЛЯ
; ОБМЕНА МАКСИМАЛЬНЫМ ЧИСЛОМ БЛОКОВ
NOP = 16. ; ЧИСЛО "ДЛИННЫХ" ОПЕРАЦИЙ
.....

2) ПРОПИСАТЬ КВАЗИДИСК НУЛЕМ
WRED: MOV #EDCSR, R3 ; АДРЕС РС ФОРМАТИРУЕМОГО КВАЗИДИСКА
; ОБМЕН МАКСИМАЛЬНЫМ ЧИСЛОМ БЛОКОВ
CLR R4
MOV #NOP, R5
WZ: MOV #<256, 264.>, R2
ZL: MOV R1, 4 (R3) ; ЗАПИСЬ НУЛЕЙ НА КВАЗИДИСК
SOB R2, ZL
SOB R5, ZL
.EXIT
    
```

Рис. 1. Структурная схема квазидиска

Рис. 2. Временные диаграммы работы схемы управления регенерацией и обменом:
 а — требование регенерации и обмена совпадают; б — требование регенерации возникает во время обмена с микроЭВМ

Рис. 3. Фрагмент программы «квазиформатирования»

```

ДРАЙВЕР ЭМУЛЯТОРА ДИСКА ( КВАЗИДИСКА )
В ОС РАФОС-2, RT-11 V.05.00
.ENABLE LC
.TITLE ED-ДРАЙВЕР КВАЗИДИСКА.

.MCALL .DRDEF,.DRDEG,.DREND,.DRFIN,.DRSET
.DRDEF ED,370,FILSTR,1024.,177720,0

; ТАБЛИЦА УСТАНОВКИ ПАРАМЕТРОВ
.DRSET CSR,160000,0.CSR,0C1

; УСТАНОВКА АДРЕСА CSR
O,CSR: MOV R0,176
MOV R0,EDEND+206
MOV R0,EDS
CMP R0,R3
RTS PC

; ОПИСАНИЕ ОБОРУДОВАНИЯ
UNIT = 4 ; МАКСИМАЛЬНОЕ ЧИСЛО УСТРОЙСТВ КВАЗИДИСКА
РЕГИСТРЫ 1- ГО КВАЗИДИСКА
CSR = 177720 ; РКС 1- ГО КВАЗИДИСКА
; ПО УМОЛЧАНИЮ

; БИТЫ CSR: ERRDAT = 40000 ; ОШИБКА ДАННЫХ-
; РY 1- ГО КВАЗИДИСКА
RCTRL = 177722 ; РY 1- ГО КВАЗИДИСКА
RD = 177724 ; РА - 1- ГО КВАЗИДИСКА ПО
; УМОЛЧАНИЮ

; РЕГИСТРЫ N- ГО КВАЗИДИСКА (N=2,3,4)
CSRN = CSR +(N-1)*10
RCTRLN = RCTRL+(N-1)*10
RDN = RD +(N-1)*10
REDN2 = 3 ; СТЕПЕНЬ 2 ОТНОСИТЕЛЬНОГО СМЕЩЕНИЯ
; РЕГИСТРОВ N- ГО КВАЗИДИСКА

; ПРОГРАММНЫЕ СОГЛАШЕНИЯ
NDERRR ОШИБКА ДАННЫХ
.DRDEB ED

;
R0 - РАБОЧИЙ РЕГИСТР
R1 - АДРЕС ТЕКУЩЕГО СЛОВА В БУФЕРЕ
R2 - СЧЕТЧИК СЛОВ
R3 - РКС КВАЗИДИСКА
R5 - УКАЗАТЕЛЬ ЭЛЕМЕНТА ОЧЕРЕДИ EDDEE

MOV EDDEE,R5 ; R5-ТЕКУЩИЙ ЭЛЕМЕНТ ОЧЕРЕДИ
TSTB QDFUNC(R5) ; СПЕЦИАЛЬНАЯ ФУНКЦИЯ ИГНОРИРУЕТСЯ
BNE EDONE

; ОПРЕДЕЛЕНИЕ НОМЕРА УСТРОЙСТВА
MOV B QNJNUM(R5),R3 ; R3-НОМЕР КВАЗИДИСКА:
BIC #176377,R3 ; НУМЕРАЦИЯ С 0 ДО 3

1X: MOV #REDN2,R0
ASL R3
SOB R0,1X ; R5-СМЕЩЕНИЕ ДЛЯ N- ГО КВАЗИДИСКА
ADD EDS,R3 ; R3-РКС N- ГО КВАЗИДИСКА

MOV QDRLKN(R5),2(R3) ; НАЧ.НОМЕР БЛОКА -> РY
MOV QBUFF(R5),R1 ; R1-АДРЕС БУФЕРА
MOV QDMCNT(R5),R2
TST R2
BNE 2X
BR EDONE ; ВЫХОД ПРИ СЧЕТЧИКЕ СЛОВ=0

2X: BPL KEADED
NEG R2 ; ПЕРЕВОД ИЗ ДОП.КОДА ДЛЯ .WRITE

; ЗАПИСЬ НА КВАЗИДИСК
MOV R2,R0 ; ЗАПИСЫВАЮТСЯ ТОЛЬКО ПОСЛЕДНИЕ БЛОКИ
BIC #0C377,R0 ; НЕПОЛНЫЕ ЗАПОЛНЯЮТСЯ НУЛЯМИ
WRITED: MOV (R1)+,4(R3)
SOB R2,WRITED
; ЗАНУЛЕНИЕ ОСТАТКА БЛОКА
WRITED: TSTB R0
BEQ EDONE
MOV R2,4(R3)
INC R0
BR WRITED

; ЧТЕНИЕ С КВАЗИДИСКА
READED: MOV 4(R3),(R1)+ ; R1 -> БУФЕР
SOB R2,READED ; R0 - СЧЕТЧИК СЛОВ
BIT ERRDAT,(R3) ; ЕСТЬ ОШИБКИ?
BEQ EDONE ; НЕТ
BIS #NDERRR,@QDCSW(R5) ; БИТ "ГРУБОЙ ОШИБКИ"

EDONE: .DRFIN ED ; ВЫХОД ИЗ ДРАЙВЕРА
EDS: .WORD EDKCSR

; НАЧАЛЬНАЯ ЗАГРУЗКА
.MCALL .DRBOT
.DRBOT ED,BOOT1,READ

; П/П ВХОДА
*EDBOOT+40
BOOT1: JMP @#BOOT-EDBOOT

```

управления (РУ) и данных (РД). Адрес начала области регистров может изменяться в пределах страницы ввода-вывода переключателями на плате квазидиска. В регистре состояния задействованы разряды 15, 14 и 13. Разряд 15 (общий признак ошибки) устанавливается, если установлен 14-й или 13-й разряды. Разряд 14 фиксирует ошибку обращения к несуществующему адресу квазидиска. Разряд 13 устанавливается, если при чтении данных зафиксирована ошибка четности. Все разряды регистра состояния доступны микроЭВМ только для чтения. Разряды сбрасываются при любой записи в регистр управления. При этом также очищаются внутренние счетчики квазидиска, подготавливая обращение к началу выбранного блока данных. В разрядах 00...09 регистра управления указывается номер блока, к которому будет производиться обращение. Регистр управления доступен микроЭВМ для записи и чтения. Размер блока информации для совместимости с ОС РАФОС — 256 слов.

Для использования квазидиска в вычислительных комплексах с ОС РАФОС разработаны следующие программные средства.

Программа квазиформатирования EDFMT SAV (рис. 3) выполняет первоначальную запись на квазидиск псевдослучайных чисел. Это необходимо для установки соответствия контрольных разрядов и содержания слова данных после включения питания.

Программа-драйвер ED.SYS (рис. 4) позволяет поблочко передавать данные (без прерывания) от 1 до 1024 блоков в одной операции обмена.

Драйвер разработан в соответствии с соглашениями ОС РАФОС на программирование внешних устройств прямого доступа с файловой структурой. ED-драйвер поддерживает до четырех устройств с номерами 0...3. В качестве системного можно использовать квазидиск с любым номером. ED-драйвер (рис. 4) позволяет выполнять настройку на конкретные физические адреса квазидиска, которые могут изменяться в зависимости от требования пользователя. Модификация адресов нулевого устройства выполняется командой монитора SET ED:CSR=ADR, где ADR — установленный физический адрес РС нулевого квазидиска. Адреса РС устройств с номером N (N=1, 2, 3) следует устанавливать равными ADR+N*10.

Для переносимости программной поддержки между микроЭВМ «Электроника 60», ДБК, «Электроника МС 1211» программы используют только основной набор команд процессора (без микропрограммного расширения).

Для загрузки с квазидиска необходимо выполнить следующие действия:
— R EDFMT; запуск программы

СИСТЕМА МОДЕЛИРОВАНИЯ НА БАЗЕ Е-СЕТЕЙ

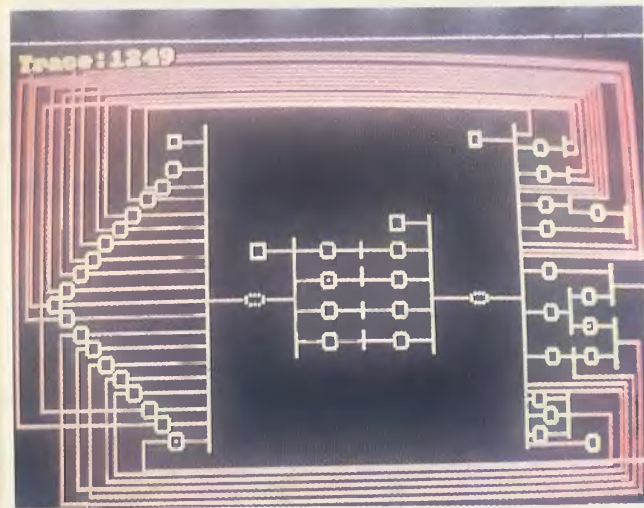
УДК 681.53.072

В. Н. Муругов

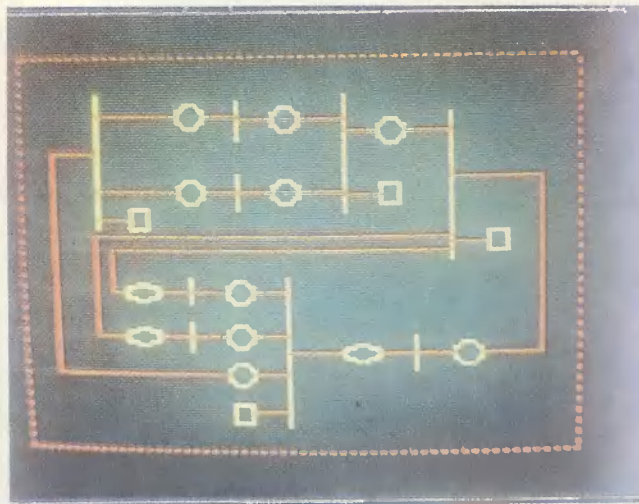
Существующие системы моделирования на базе Е-сетей реализованы на больших и мини-ЭВМ, но не используют графическое представление при интерпретации на ЭВМ.

Разработанная система имитационного моделирования предназначена для работы на ПЭВМ ЕС 1841 и микроЭВМ, совместимых с IBM PC/XT, имеющих цветной графический дисплей с CGA или EGA адаптером.

Система моделирования включает в себя:
цветной графический редактор для создания и редактирования структуры Е-сети, описывающей моделируемую систему;
специальный язык программирования для описания функционирования Е-сети;



Изображение Е-сети в графическом редакторе



Отладка созданной модели в графическом режиме

средства генерации исполняемого файла модели;

графический отладчик Е-сетей моделей.

Предусмотрена возможность автоматического накопления статистики по различным параметрам модели, создания удобного интерфейса между пользователем и моделью. Объем ПО составляет 500 Кбайт дисковой памяти.

Предоставляемый пользователю инструмент моделирования может применяться для имитационного моделирования вычислительных структур и других дискретных параллельных систем.

Телефон 248-82-07, Москва

РЕКЛАМА

ГПНТБ СО АН СССР проводит для организаций и предприятий разработку, передачу готовых программных продуктов, обучение специалистов, консультации и экспертизы в областях информатики, компьютеризации различных процессов, связанных с документальными потоками (документы, факты, концепции). В частности, разработаны электронные картотеки, позволяющие работать специалисту, работнику НТИ, эксперту, потребителю в режиме автоматизированного рабочего места (АРМ).

По желанию заказчиков создаются проблемно-ориентированные базы данных. Проводится обучение в

компьютерном классе основам информатики, работе на персональных компьютерах и другим дисциплинам. Созданы и ведутся оригинальные базы данных «Автореферат», «Конференции», «Периодика» и др. Эти БД могут передаваться полностью или фрагментами по выбранному заказчиком профилю на магнитных лентах или дискетах. Осуществляется передача информационных технологий на ЕС ЭВМ, СМ ЭВМ, персональных компьютерах.

Адрес для справок: 630200, Новосибирск-200, ул. Восход 15, ГПНТБ СО АН СССР, отдел автоматизации, тел. 66-75-79

НАШ ОПЫТ—ЭТО ВАШ УСПЕХ!

Что может связывать разработчика электронной аппаратуры, архитектора, конструктора и модельера? **PROSYSTEM Computer Technic** утверждает — **СРЕДСТВА ПРОЕКТИРОВАНИЯ**.

Предлагаемые **PROSYSTEM** аппаратные и программные средства:

— CAD/CAE разработки электронных схем,

— CAD/CAE/CAM/CIM разработки печатных плат,

— CAD/CAM/CIM моделирования и конструирования деталей машин и механизмов,

— CAD/CAE/CAM проектирования строительных объектов и сооружений,

— CAD/CAM проектирования фактуры тканей,

а также оригинальная концепция высокоскоростных сетевых архитектур, разработанная специалистами

PROSYSTEM, позволит Вам резко увеличить производительность Вашего «Центра» проектирования и «Службы» управления производством, «сердцем» которых может стать любой из поставляемых **PROSYSTEM** персональных компьютеров — от недорогих 16-разрядных ПК, совместимых с IBM PC/XT или IBM PC/AT, до высокопроизводительных, полностью 32-разрядных ПК, приближающихся по производительности к большим компьютерам.

Мощные графические станции с высоким разрешением (до 1280×1024) на базе 16-ти и 32-разрядных ПК, матричные и лазерные принтеры, рулонные и планшетные плоттеры форматов A0-A3, сверлильные и фрезерные настольные станки, сканеры — вот далеко неполный перечень периферийных устройств, предлагаемых **PROSYSTEM**, из которых Вы можете самостоятельно или с нашей помощью сформировать систему произвольной сложности.

Не удивляйтесь, если Вы не нашли в этой рекламе устройств внешней памяти. Вопреки установившемуся мнению о необходимости оборудовать каждый компьютер винчестерским накопителем большой емкости, мы утверждаем, что в большинстве случаев в этом нет нужды. Высокоскоростная локальная сеть, совместимая по протоколу обмена данными с **ETHERNET**, с центральным устройством обслуживания файлов (**FILE SERVER**) не только сэкономит Ваши денежные средства, но обеспечит надежную защиту данных и авторизованный доступ к информации. Но если Вы при-



Рабочее место инженера — проектировщика

Фотоплоттер BEIE-P40



держиваетесь традиционных взглядов, либо Вас интересуют автономные рабочие станции, **PROSYSTEM** может оснастить их устройствами внешней памяти емкостью от 20 до 400 МВ. Разрабатывая для Вас кон-



печатных плат на московском заводе «Энергоприбор»

2. Мощный высоконадежный планшетный фотоплоттер BEIE-P40 с вертикальным расположением стола и занимающий минимум площади:

- рабочее поле — 610×610 мм,
- стандартная толщина фотоматериала — 0,18 мм (пленка), 3,3 мм (стекло),
- разрешающая способность — 0,00075 мм,
- скорость рисования — 150 мм/с,
- входные данные — EIE формат, P-274 GERBER формат.

3. Компактный, дешевый, высокоточный планшетный плоттер BMP-3200 с возможностью работы в вертикальном положении:

- формат — A3 (404×285 мм),
- скорость вычерчивания линий и дуг — 40 см/с,
- точность — 0,025 мм,
- количество перьев — 8,
- входной язык — GP-GL и HP-GL эмуляция.

4. Крупноформатный рулонный плоттер GP-1002(A0), GP-1102(A1), занимающий незначительную площадь рядом с рабочим местом:

- формат — A0 (2000×864 мм), A1 (2000×594 мм),
- скорость вычерчивания линий и дуг — 80 см/с,
- точность — 0,005 мм,
- количество перьев — 8,
- входной язык — GP-GL и HP-GL эмуляция.

фигурацию системы «под ключ», **PROSYSTEM** уделяет особое внимание не только обеспечению ее функциональной полноты, производительности и удобству эксплуатации, но и повышенной эффективности использования оборудования при решении конкретных задач и рациональной организации труда высококвалифицированных специалистов.

1. Мощный высоконадежный барабанный фотоплоттер BEIE-P30:

- рабочее поле — 610×914 мм,
- стандартная толщина фотоматериала — 0,18 мм,
- разрешающая способность — 0,00075 мм,
- скорость рисования — 150 мм/с,
- входные данные — EIE формат, P-274 GERBER формат.



PROSYSTEM
Computer Technic

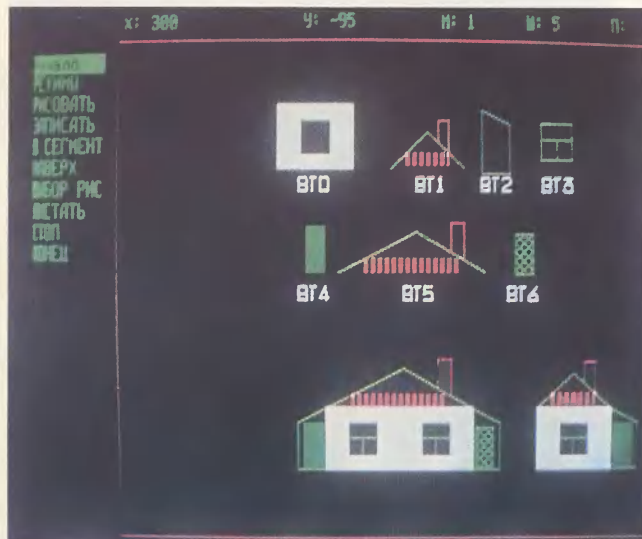
Австрия
Austria, Parkring
12 A/8/5,
A-1010, Vienna
Tel. 51 34 760-0
Tlx. 112937
Fax 51 34 764

СССР
СССР 123022 Москва,
ул. Трехгорный вал,
д. 2, строение 2
Тел. 255-01-01
Телекс: 411941
Факс: 230-22-92

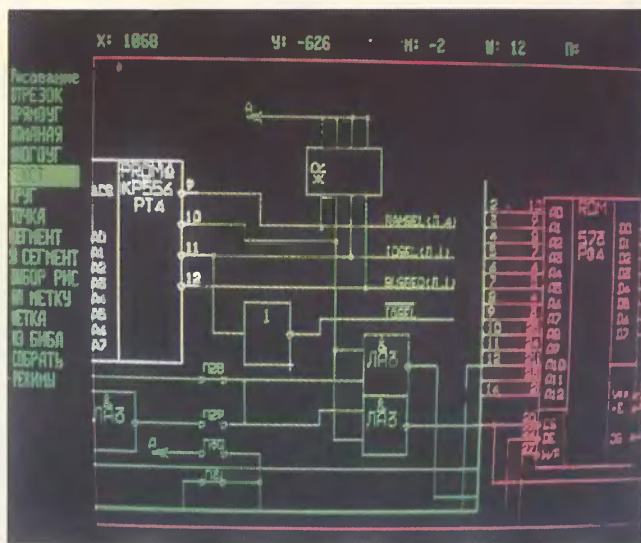
ИНТЕРАКТИВНАЯ ГРАФИЧЕСКАЯ СИСТЕМА ГРАФ

(К ст. Г. И. Алексеева, С. П. Мильникова)

Интерактивная графическая система ГРАФ предназначена для создания в режиме диалога принципиальных электрических схем, демонстрационных чертежей и рисунков двумерных объектов, а также их хранения, редакти-



Пример построения рисунка из библиотечных элементов



Редактирование принципиальной электронной схемы

рования и вывода на принтеры различных типов. Система ГРАФ разрабатывалась как одно из средств повышения уровня автоматизации проектирования и документирования программных и технических систем.

ЛОГИЧЕСКИЙ АНАЛИЗАТОР

(К ст. Андреева Б. М., Леухина С. П.)

Логический анализатор обеспечивает следующие возможности по регистрации, отображению и обработке цифровой информации об изменении логических состояний контролируемой схемы:

- запуск процесса регистрации по 1-му или $(n+1)$ -му обнаружению в контролируемом потоке данных кодового слова запуска ($n=1...9$);

- регистрацию данных, предшествующих кодовому слову запуска на заданное число тактов записи;

- регистрацию данных в режиме сравнения массивов, при этом в одном цикле записи процесс регистрации запускается дважды по двум последовательным или $(n+1)$ -м обнаружениям в контролируемом потоке данных кодового слова запуска;

- регистрацию данных в перечисленных режимах по фиксированным значениям тактов внутренней или внешней синхронизации;

- отображение зарегистрированной информации в виде временных диаграмм;

- возможность растяжки и сжатия отображаемой информации;
- возможность попеременного отображения соответствующих участков двух массивов, зарегистрированных в режиме сравнения;
- отображение служебной информа-

- ции о параметрах выбранных режимов (записи и отображения);

- вычисление расстояния между курсорами в тактах записи и отображение вычисленной величины на экране;

- задание и поиск в массиве зарегистрированной информации новых квалифицирующих слов.



квазиформатирования
 — INI/BAD/NOQ EDN;: инициализация квазидиска
 — COPY/SYS/*.*EDN;: копирование требуемых файлов на квазидиск
 — COPY/BOOT RT11SJ.SYS EDN;: копирование системного загрузчика
 — BOOT EDN;: загрузка ОС с квазидиска.

Для включения драйвера в ОС необходимо выполнить команды:

```
MACRO ED
LINK/EXE: SY:ED.SYS ED:
INSTALL ED:
```

Конструктивно квазидиск выполнен на двухслойной печатной плате микроЭВМ «Электроника 60» (280×240 мм). Общее количество установленных на плате ИС — 122, ИС ОЗУ — 72. Незначительная доработка квазидиска позволила заменить ИС K565PY5 на ИС K565PY7 при соответствующем увеличении емкости квазидиска либо уменьшении его габаритных размеров.

В настоящее время разработан и испытан опытный образец квазидиска. Его эксплуатация в составе микроЭВМ «Электроника 60.1» и ДВК2М в среде РАФОС-2, RT-11 V.05. полностью подтвердила правильность принятых технических решений.

Телефон 227-10-30 (с 9 до 17), Киев, Васильев В. Г.

УДК 681.326—181.4

Ю. Н. Котельников, А. М. Крочев, А. С. Куликов, С. П. Никитин, Е. А. Обходов

ТЕХНОЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ СЕРИИ УТК6

На базе ОЭВМ K1816* разработана и готовится к промышленному выпуску серия одноплатных универсальных технологических контроллеров УТК6: УТК6А, УТК6Д, УТК6М, УТК6С (рис. 1). Они предназначены для управления технологическими объектами различной сложности — от простых агрегатов до сложных автоматических линий (в последнем случае используется группа контроллеров, объединенных общей системой связи).

Все модификации УТК6 имеют внешний канал типа ИРПС (интерфейс радиальной последовательной связи) для обмена информацией с дисплеем или другими ЭВМ.

Основные функциональные узлы контроллера УТК6А (рис. 2):

микроЭВМ K1816 со схемами расширения памяти;

формирования управляющих сигналов и внутренней магистрали контроллера с узлом оптоэлектронной развязки;

* Кобылинский А. В., Линовецкий Г. П. Однокристалльные микроЭВМ серии K1816 // Микропроцессорные средства и системы. — 1986. — № 1, С. 10—19.

```

P/N ЧТЕНИЯ ЗАГРУЗЧИКА
RO - НОМЕР БЛОКА
R1 - ЧИСЛО СЛОВ ДЛЯ СЧИТЫВАНИЯ
R2 - АДРЕС БУФЕРА
R3 - АДРЕС ПК С СИСТЕМНОГО УСТРОЙСТВА

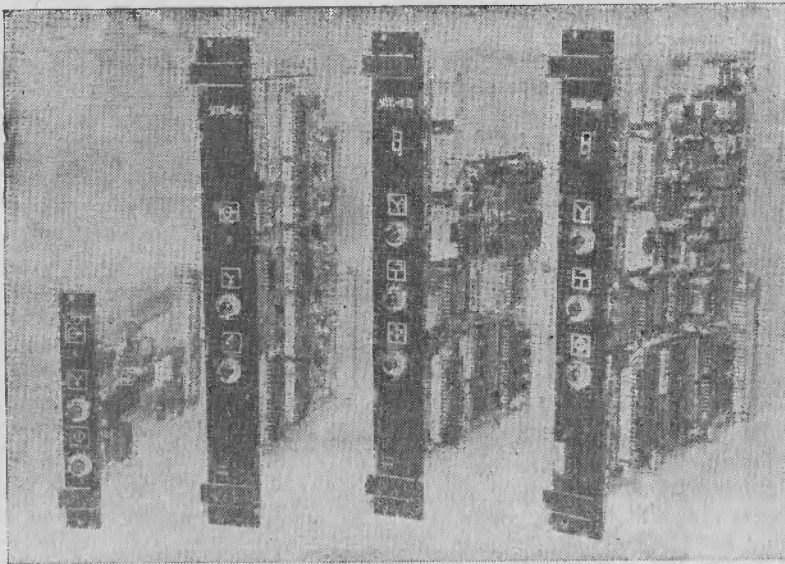
.=EDBOOT+206
EDSSET: .WORD EDHCSR ; ПК В ДРАЙВЕРЕ ED
.=EDBOOT+210
READ: MOV EDHCSR,R3 ; R3-АДРЕС ПК СИСТ. КВАЗИДИСКА
      MOV RO,R3 ; НАЧ.НОМЕР БЛОКА -> РУ
READBT: MOV 4(R3),(R2)+ ; ЧТЕНИЕ БЛОКОВ ВТОРИЧНОГО ЗАГРУЗЧИКА,
      SOB R1,READBT ; МОНИТОРА, ДРАЙВЕРА
      BIT #ERRDAT,(R3) ; ЕСТЬ ОШИБКИ?
      BNE B.ERR ; ДА
      RTS PC ; НЕТ:ВОЗВРАТ
B.ERR: JMP BIOERR ; ДА:=> НА ОБРАБОТКУ ОШИБОК
EDNU: .WORD 0 ; НОМЕР СИСТЕМНОГО КВАЗИДИСКА
;
; ВЫЧИСЛЕНИЕ АДРЕСА СИСТЕМНОГО КВАЗИДИСКА
DEFNU: MOV RO,EDNU ; RO-НОМЕР СИСТЕМНОГО УСТРОЙСТВА
SHLNU: MOV #REDN2,R1
      ASL RO
      SOB R1,SHLNU
      ADD EDHCSR,RO
      MOV RO,EDSSET ; АДРЕС ПК СИСТЕМНОГО КВАЗИДИСКА
      RTS PC
;
ПРОГРАММНЫЙ ЗАГРУЗЧИК
.=EDBOOT+574
BOOT: MOV #10000,SP
      JSR PC,DEFNU ; ОПРЕДЕЛИТЬ АДРЕС СИСТЕМНОГО УСТРОЙСТВА
      MOV #TIMRTI,@#100 ; TIMRTI-АДРЕС ТАЙМЕРА
      MOV #2,RO ; НАЧАЛО ВТОРИЧНОГО ЗАГРУЗЧИКА С 2- ГО БЛОКА
      MOV #<256.*4>,R1 ; ДЛИНА ВТОРИЧНОГО ЗАГРУЗЧИКА В СЛОВАХ
      MOV #1000,R2 ; АДРЕС ЗАГРУЗКИ
      CALL READ
      MOV #READ-EDBOOT,@#VREAD ; ЗАПОЛНЕНИЕ СИСТ. ИНТЕРФЕЙСНЫХ ПОЛЕЙ
      MOV #VBDNAM,@#VBDNAM
      MOV EDNU,@#VBDDEVU ; НОМЕР КВАЗИДИСКА
      JMP #VBDVBOOT ; => ВО ВТОРИЧНОМ ДРАЙВЕРЕ
TIMRTI: RTI #VBDVBOOT ; БЛОКИРОВКА ПРЕРЫВАНИИ ТИ ТАЙМЕРА

.DREND ED
.END
  
```

Рис. 4. Текст драйвера квазидиска

Основные технические данные

Параметр	УТК6А	УТК6Д	УТК6С	УТК6М
Быстродействие ЭВМ тыс. опер./с	400	400	400	400
Объем, байт				
ОЗУ	до 320	до 320	до 320	64
ПЗУ	4096	4096	4096	4096
Число каналов:				
аналогового контроллера	16	—	—	8
аналогового управления дискретного контроля	2	—	—	—
дискретного управления	8	24	—	8
внешней связи (типа ИРПС)	16	16	—	12
общее	1	1	5	1
Точность преобразования аналоговых сигналов в диапазоне 0 ... +10В, мВ	42	40	—	28
Габаритные размеры мм	$30 \times 262 \times 1200$	$30 \times 262 \times 245$	$30 \times 262 \times 245$	$20 \times 128,7 \times 245$
Цена (ориентировочная), руб.	1200	1000	1000	600

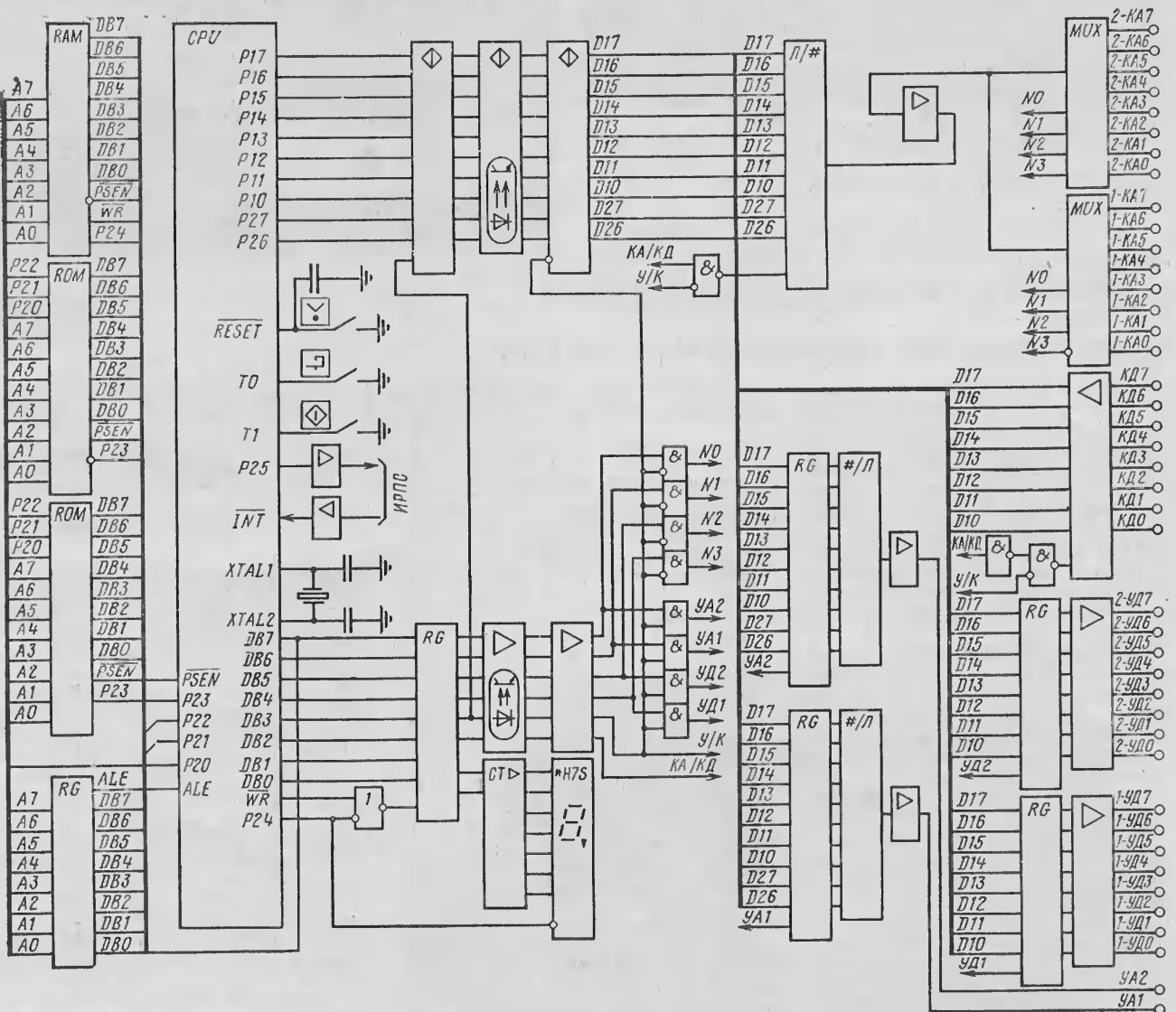


аналогового ввода-вывода, дискретного ввода-вывода, последовательного обмена, управления и индикации.

В контроллере использована модификация микроЭВМ К1816ВЕ035, которая не содержит на кристалле постоянного запоминающего устройства (ПЗУ), а объем оперативной памяти данных ограничен 64 ячейками. Поэтому в схеме предусмотрена возможность подключения внешнего ПЗУ объемом до 4096 байт и расширения памяти данных на 256 байт.

Рис. 1. Контроллеры серии УТК6

Рис. 2. Структурная схема контроллера УТК6А



Внешняя постоянная память программ реализована в двух микросхемах с ультрафиолетовым стиранием информации КР573РФ2 по 2048 байт каждая. Адрес любой ячейки памяти в пределах 4096 байт определяется 12-разрядным словом, восемь разрядов формируются по шине данных на адресном регистре, а четыре — младшими разрядами P20...P23 порта 2.

В зависимости от состояния разряда P23 разрешается выбор кристалла одной из микросхем памяти. Первая микросхема, как правило, содержит стандартные программы, вторая поставляется незаполненной и предназначена для программ пользователя.

Считывание и запись информации во внешнюю память данных (микросхема КР537РУ8) осуществляется стандартными командами обращения при наличии низкого уровня по шине P24.

Для организации подсистем аналогового и дискретного ввода-вывода в контроллере используется внутренняя двунаправленная 10-разрядная магистраль, формируемая разрядами P17...P10, P27, P26 портов 1 и 2 ОЭВМ. Вид и направление приема-передачи по магистрали определяются управляющими сигналами, фиксируемыми на выходах регистра управления, который загружается по шине данных при наличии высокого уровня P24.

Все магистральные шины и сигналы управления, связанные с подсистемами аналогового и дискретного ввода-вывода, разделены гальванически с помощью узла оптоэлектронной развязки.

Устройство аналогового ввода содержит аналоговый мультиплексор на 16 входов, буферный усилитель и 10-разрядный аналого-цифровой преобразователь. Входное сопротивление по каждому каналу не менее 400 МОм.

В состав устройства аналогового вывода входят два 10-разрядных регистра, загружаемых по магистрали контроллера, два 10-разрядных цифро-аналоговых преобразователя и два выходных усилителя с допустимым током нагрузки по выходам 5 мА. Диапазон изменения аналоговых сигналов 0...10 В.

Устройство дискретного ввода выполнено на микросхеме восьмиканального буферного усилителя. В режиме ввода микросхема доступна для считывания информации в ЭВМ между циклами аналого-цифрового преобразования. Уровни входных сигналов: Лог. 0 — не более 0,4 В, Лог. 1 — не менее 2,4 В.

Устройство дискретного вывода объединяет две группы по восемь каналов управления, состоит из двух регистров, подключаемых к магистрали, и усилителей. Выходные ключи усилителей имеют открытый коллек-

торный выход и допускают ток в замкнутом состоянии до 100 мА. Напряжение в коммутируемой цепи при разомкнутых ключах — 30 В.

Посылки по линии передачи данных формируются разрядом P25 ОЭВМ, ответные посылки на линии приема данных поступают на вход INT. Прием и передача в стандартном коде организуются программно.

С помощью кнопочных переключателей передней панели осуществляется начальная установка ОЭВМ (при наличии постоянного запоминающего устройства выход на начальный адрес программы контроллера) и пуск контроллера в работу или в режим автоматической проверки работоспособности.

На семисегментном цифровом индикаторе индицируется номер этапа технологического процесса (от 0 до 9) и сообщения об отклонениях в ходе процесса или при тест-проверке (мигающие сигналы с чередованием цифр).

Контроллер питается напряжением

ми: +5 В — изолированное для ОЭВМ, +5 В, +15 В, -15 В. Потребляемая мощность не превышает 10 В·А.

Фиксированная частота (4608 кГц) кварцевого резонатора обеспечивает синхронизацию работы ОЭВМ и является опорной при формировании импульсов канала ИРПС.

В контроллере УТК6Д отсутствуют каналы аналогового ввода и вывода, однако за счет использования мультиплексора для передачи дискретных сигналов расширена подсистема дискретного ввода до 24 каналов. Остальные характеристики полностью соответствуют контроллеру УТК6А.

Контроллер УТК6М (рис. 3) — малогабаритный вариант серии УТК6А, в котором не предусмотрены расширение оперативной памяти данных и вывод аналоговой информации; вывод дискретной информации сокращен до 12 каналов. Остальные функции сохранены.

Регистр управления вырабатывает сигналы выбора адреса аналогового мультиплексора и четыре управляю-

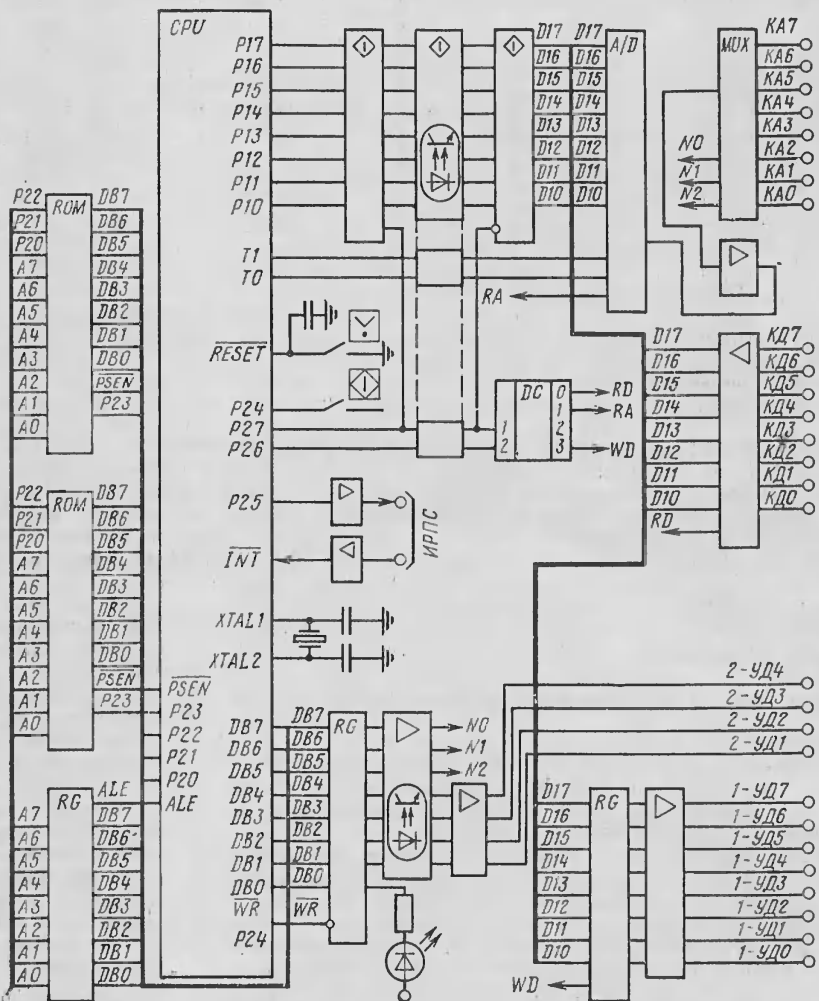


Рис. 3. Структурная схема контроллера УТК6М

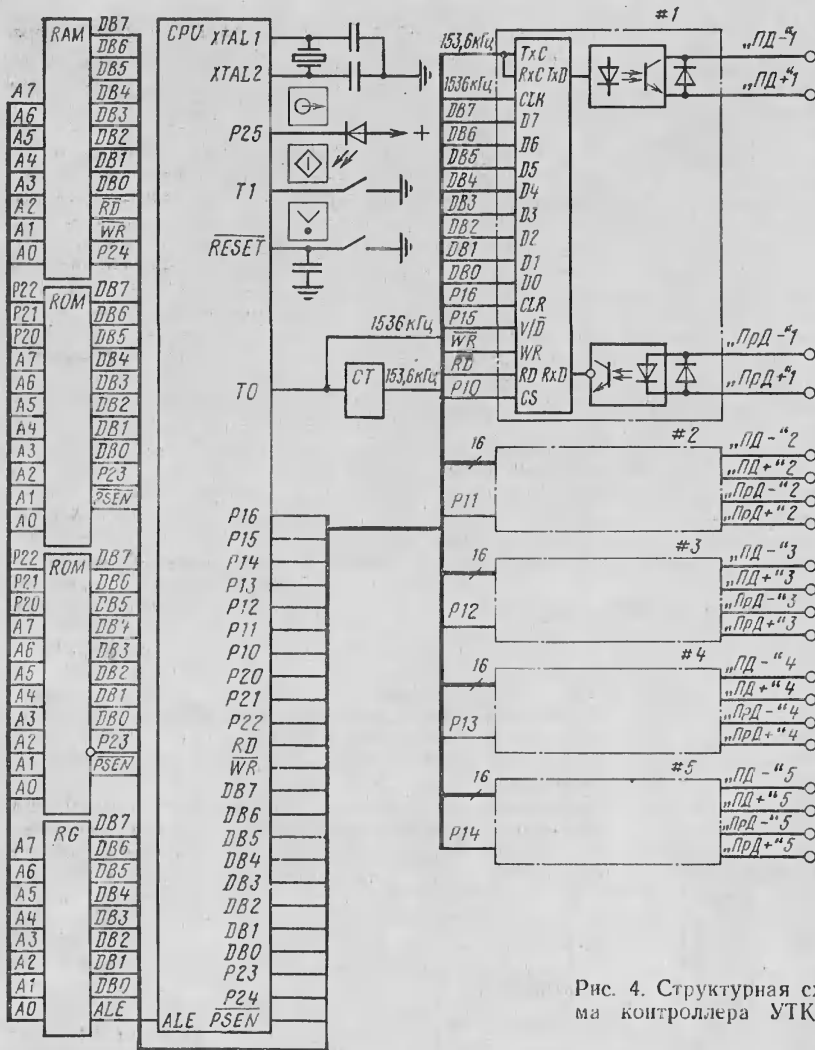


Рис. 4. Структурная схема контроллера УТК6С

щих сигнала дискретного вывода. Младший разряд регистра задает работу индикатора на передней панели.

Разряды P27, P26 порта 2 ОЭВМ определяют направление и характер работы внутренней восьмиразрядной

магистральной контроллера D17...D10. Сигналы с выхода дешифратора управляют устройствами дискретного ввода и вывода, аналогового ввода. Два младших разряда аналого-цифрового преобразователя связаны с

тестируемыми входами ОЭВМ, которые работают только на прием информации. Кнопки начальной установки (управляется по входу RESET) и запуска (P24) расположены на передней панели и выводят контроллер на рабочий режим.

Основное назначение контроллера УТК6С (рис. 4) — связь между контроллерами при модульном построении системы управления и обмен информацией с ЭВМ верхнего уровня или выносным дисплеем. УТК6С работает в режиме асинхронного обмена информацией. Программа, содержащаяся в ПЗУ контроллера, обеспечивает работу пяти каналов последовательной связи.

Основной элемент каждого из пяти каналов — микросхема КР580ИК51. На ее входы подаются высокочастотные тактовые импульсы (CLK), сигнал установки в начальное состояние (CLR), тактовые импульсы для синхронизации данных (TxC и RxC) и передаваемые данные. В режиме асинхронной передачи импульсы синхронизации вырабатываются ОЭВМ. При частоте кварцевого резонатора 4608 кГц в результате деления ОЭВМ счетчиком и микросхемой КР580ИК51 (организуется программно) получается частота синхроимпульсов приема и передачи, соответствующая скорости передачи данных 9600 Бод.

Для организации внутреннего режима работы КР580ИК51 используется разряд P15. В зависимости от его состояния (вход V/D) через шину DB0...DB7 передается либо управляющее слово, либо данные.

Работа с каждой из пяти микросхем происходит последовательно поочередным включением в Лог.0 разрядов P10...P14 ОЭВМ (входы выбора кристалла микросхемы КР580ИК51).

Выход передатчика (TxD) и вход приемника (RxD) разделены гальванически от линий передачи данных оптоэлектронными схемами.

Телефон: 127-98-68, Москва

Статья поступила 03.02.88

УДК 681.3.022

Е. А. Ходаковский

ПРОСТОЙ ТЕРМИНАЛ МИКРОЭВМ

Предлагаемый терминал может быть использован в качестве одного из устройств, работающих совместно с процессором ввода-вывода (ПВВ), построенных на базе МП КР580ВМ80А.

Терминал содержит большое количество микросхем, обеспечивает обслуживание клавиатуры (64 клавиши) и дисплея (8 семисегментных индикаторов). Клавиатура и дисплей связаны с ПВВ через параллельный интерфейс (БИС КР580ВВ55), порт РВ которого настроен на ввод, а остальные порты — на вывод информации. Две микросхемы К589АП26 используются как буферы сопряжения параллельного интерфейса с шиной ПВВ и входами семисегментного индикатора соответственно.

При ненагруженной шине ПВВ буферы могут отсутствовать. Селектор адреса выполнен на микросхеме К155ЛА2 и настроен на адрес порта 00Н...03Н.

Через порт РС программа сканирует клавиатуру безгудим нулем, а через порт РВ процессор принимает информацию о нажатой клавише. Дребезг контактов клавиатуры подавляется программно. Одновременно через порт РС программа обеспечивает динамическую индикацию информации, непрерывно подкачиваемой микропроцессором в порт РА. Индицируемая информация тоже дешифруется в коды семисегментного индикатора тоже программно.

Данный терминал можно использовать не только совместно с ПВВ, но и при построении недорогих МП-систем, например, домашнего компьютера минимальной конфигурации.

Телефон 433-41-94 (с 20 до 22), Киев

Сообщение поступило 21.01.87

В. А. Корнев, Э. М. Пройдаков, Н. Н. Щелкунов,
А. П. Дианов

УНИВЕРСАЛЬНЫЙ КОНТРОЛЛЕР НА БАЗЕ ОЭВМ КР1816ВЕ51

Контроллер предназначен для применения в медицинских приборах и аппаратуре, для съема и обсчета медицинских кривых, а также в системах автоматизации научного эксперимента и АСУ ТП (см. рисунок).

Выход контроллера на стандартную шину ИА1 позволяет использовать этот контроллер в микроЭВМ типа СМ1800, СМ1810, а с помощью последовательного интерфейса — подсоединять периферийные устройства к стандартному выходу на ИРТС или RS-232, а также включать контроллер в средне- и низкоскоростные локальные сети.

Технические характеристики контроллера:

ОЗУ, Кбайт	4
РПЗУ, Кбайт	32
Число разрядов:	
АЦП	10
ЦАП	9
Число аналоговых входов	8
выходов	8
Время преобразования по одному каналу мкс	200
Потребляемая мощность, Вт	14
Формат плат	Е2

Однокристалльная микроЭВМ КР1816ВЕ51 (DD2) содержит в одном корпусе 8-разрядное ЦПУ, 4096 байт РПЗУ, 128 байт ОЗУ, часть которого используется под четыре банка по восемь регистров в каждом, четыре 8-разрядных порта ввода-вывода, высокоскоростной программируемый последовательный порт, два 16-разрядных таймера-счетчика и другие устройства. Эта микроЭВМ имеет более развитый по сравнению с микроЭВМ К1816ВЕ48 набор команд, возможность адресации 124 Кбайт памяти (60К РПЗУ — память программ и 64 Кбайт ОЗУ — память данных), расположенной вне кристалла (на плате), 128 доступных пользователю программно-управляемых флагов, двухуровневую структуру приоритетного прерывания и другие возможности [2...4]. Питание осуществляется от одного источника напряжения +5 В. С помощью конвертора CONV51 [5] программы, разработанные для однокристалльной микроЭВМ К1816ВЕ48, можно перенести на КР1816ВЕ51.

В контроллере порт P0 реализует мультиплексорную шину адрес-данные и работает как возбудитель с тремя состояниями. В начале цикла внешней памяти младшие восемь битов адреса включаются входными сигналами на P0. Сигнал ALE (разрешение фиксации адреса) подается на вход STB (strobe) регистра DD7, который производит их дальнейшую выборку на шину адреса. Старшие восемь разрядов адреса формируются портом P2 при выполнении в программе команды MOVX (обращение к внешней памяти). Данные формируются портом P0. Регистры DD6, DD7 усиливают адресную шину. Сигнал управления OE (Output Enable) заземлен и адресные сигналы передаются всегда на выход микросхем, так как шина адреса однонаправленная. Шина данных усиливается буфером DD19.

DD13, DD16 — микросхемы РПЗУ (8 Кбайт×7) с ультрафиолетовым стиранием, DD17, DD18 — ОЗУ (2 Кбайт×8). Внутренняя память ОЭВМ (4 Кбайт РПЗУ и 128 байт ОЗУ) используется для программ инициализации и тестирования контроллера. Внешняя память РПЗУ и ОЗУ наращивается до 60 и 64 Кбайт соответственно независимо друг от друга с теми же самыми адресами, A0...A12 для РПЗУ и A0...A10 для ОЗУ. Сигналы шины данных D0...D7 для всех микросхем памяти DD13...DD18 одинаковы. Постоянная память подключается на шину внешних данных с по-

мощью сигнала PSEN (Program Store Enable — разрешающие программной памяти). Режим записи производится в ОЗУ по выходному сигналу WR. Инверторы DD5.3...DD5.6, DD3.4, DD3.5; ЗИЛИ-НЕ DD9.1, DD9.2; 2И-НЕ DD10.1, DD10.2 и емкости C4, C5 служат для согласования временных характеристик сигналов WR и CS (выбор кристалла), подающихся на выходы CE1, CE2 оперативной памяти. Дешифрация выбора кристаллов осуществляется тремя старшими линиями адреса A14, A16, поступающими на входы D0...D2 дешифратора DD8. Сигнал CS, сформированный инверторами DD3.1, DD3.2, DD3.6, DD20.1...DD20.3 и 8И-НЕ DD4, DD31, DD32 и адресами A1...A5, осуществляет выбор микросхем DD11, DD24, DD25.

Контроллер клавиатуры и дисплея DD11 поддерживает функциональную клавиатуру (64 клавиши) и индикацию в виде 16-ричного знакоцифрового индикатора. С помощью программируемого параллельного интерфейса DD31 к контроллеру подключается аналого-цифровой (DA2) и цифро-аналоговый (DA3) преобразователи. Для коммутации аналоговых входных сигналов ±5 В на АЦП (К1113ПБ1), а также выходных сигналов с ЦАП используются коммутаторы напряжений DA1, DA8. В качестве цифро-аналогового преобразователя используется микросхема 10-разрядного умножающего ЦАП К572ПА1, выходы которого подключены к выводам портов PBO...PB7; PC1. А на 15-й вход ЦАП подается опорное напряжение, формируемое ОУ DA5. Устройство выборки хранения реализуется на ОУ A9...A16.

В программном параллельном интерфейсе DD32 порты PA и PB буферизуются регистрами DD33 и DD34 соответственно. С их помощью возможно управление цифрочечью, электродами и др.

Программное обеспечение контроллера разработано с помощью кросс-системы на языках программирования ПЛ/М-51 [6] и ASM-51 [7] под управлением MS-DOS в среде программного эмулятора ISIS-II. На ПЭВМ типа IBM PC XT и отладочной плате SDK51. Аналогичный по функциям пакет программ «Кросс-51» имеется на СМ 1800/СМ 1810.

Программное обеспечение контроллера включает: начальный тест; отладочный монитор (типа SDK-51 [8]); программы обмена данными (по последовательному интерфейсу, вводу с клавиатуры, работы с устройством алфавитно-цифровой индикации или дисплеем, вводом-выводом аналоговой информации); прикладные программы съема и обработки медицинских кривых.

При решении асинхронных задач (до восьми) для их диспетчеризации в контроллере может использоваться ядро ОС RMX-51 [9], по функциям аналогичная известным ОС RMX-80 и RMX-88. Однако в RMX-51 существенно улучшена диспетчеризация сообщений, которая учитывает их проходимость по локальной сети микроконтроллеров, прямую адресацию узлу локальной сети и задаче, естественное определение типа события (прерывание, истечение заданного интервала времени, получение сообщения) без явного привлечения механизма обменников.

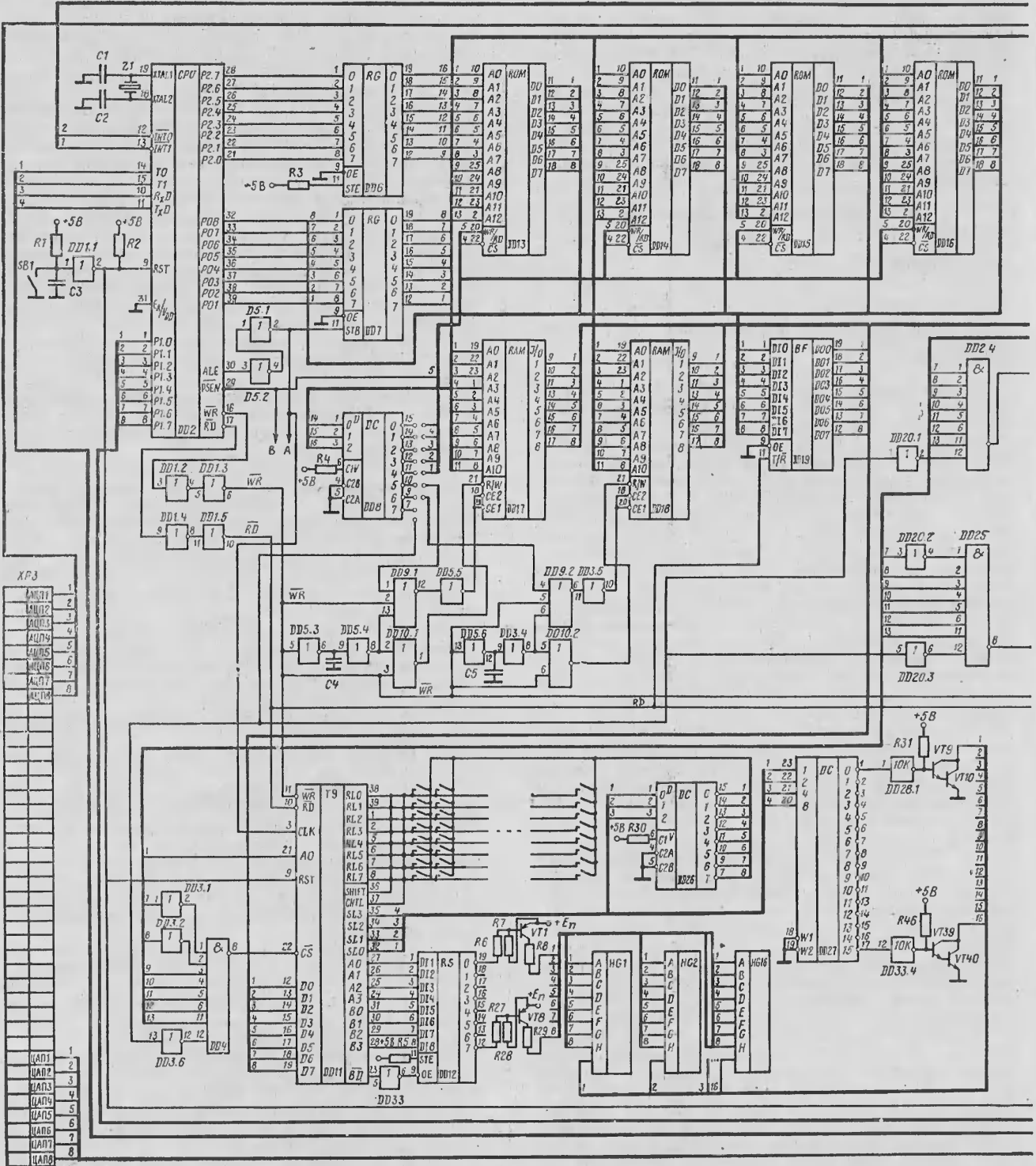
129301, Москва, ул. Касаткина, д. 3, ВНИИМТ,
тел. 283-97-64

ЛИТЕРАТУРА

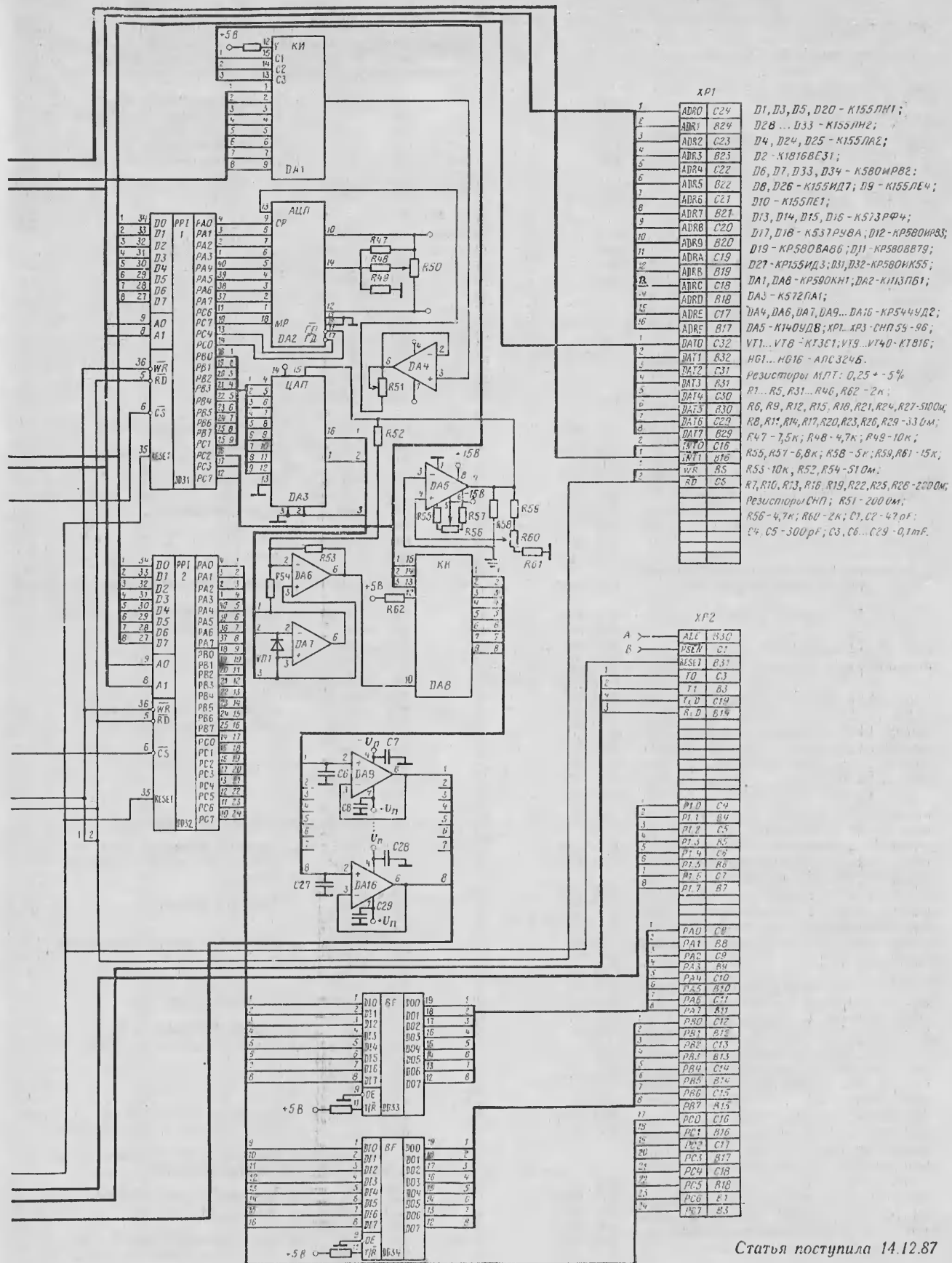
1. Крылов Е. И. Однокристалльные микроЭВМ серий К1814, К1820, К1816 // Микропроцессорные средства и системы. — 1985. — № 2. — С. 3—7.
2. Microcontroller Handbook // Intel Corp. — 1986. — N 210918.
3. Весноватов М. Г., Карацуба Г. В., Павлов В. В., Старшина В. А. Перспективные однокристалльные ЭВМ // Микропроцессорные средства и системы. — 1987. — № 2. — С. 7—8.
4. Guide to using Distributed Control Modules // Intel Corp. — 1984. — N 146312.

5. M C S-51 8048 to 8051 Assembly Languages Converter Operating Instruction for ISIS-II//Intel Corp.— 1979.— N 9800934.
6. P L/H-51 User's Guide. USA.— Intel Corp.— N 121966-002.

7. A S M-51 Macro Assembler User's Guide//Intel Corp.— 1983.— N 9800937.
8. S D K-51 Monitor Listing Manual.— Intel Corp.— 1981.— N 121590.
9. Операционные системы реального времени для микроЭВМ.— М.: МЦНТИ/МНИИПУ, 1984.



Принципиальная схема контроллера



XP1

1	ADRO	C24	D1, D3, D5, D20 - K155ЛН1;
2	ADR1	B24	D28... D33 - K155ЛН2;
3	ADR2	C23	D4, D24, D25 - K155ЛН2;
4	ADR3	B23	D2 - K1816ВС31;
5	ADR4	C22	D6, D7, D33, D34 - K580MPB2;
6	ADR5	B22	D8, D26 - K155ЛН7; D9 - K155ЛН4;
7	ADR6	C21	D10 - K155ЛН1;
8	ADR7	B21	D13, D14, D15, D16 - K573PФ4;
9	ADR8	C20	D17, D18 - K537PЧ8A; D12 - KP580MPB3;
10	ADR9	B20	D19 - KP580BAВ6; D11 - KP580BВ79;
11	ADRA	C19	D27 - KP155ИД3; D31, D32 - KP580ИК55;
12	ADRБ	B19	DA1, DA8 - KP590KH1; DA2 - KИ13ПБ1;
13	ADRC	C18	DA3 - K572ПA1;
14	ADRD	B18	DA4, DA6, DA7, DA9... DA16 - KP544УД2;
15	ADRE	C17	DA5 - K140УД8; XP1. KP3 - ЧП55-96;
16	ADRF	B17	VT1... VT8 - KТ3С1; VT9 - VT40 - KТ816;
17	DA10	C32	HГ1... HГ16 - АПС324Б.
18	DA11	B32	Резисторы МЛТ: 0,25 + - 5%
19	DA12	C31	R1... R5, R31... R46, R62 - 2к;
20	DA13	B31	R6, R9, R12, R15, R18, R21, R24, R27 - 510Ом;
21	DA14	C30	R8, R11, R14, R17, R20, R23, R26, R29 - 33 Ом;
22	DA15	B30	R47 - 7,5к; R48 - 4,7к; R49 - 10к;
23	DA16	C29	R55, R57 - 6,8к; R58 - 5к; R59, R61 - 15к;
24	DA17	B29	R53 - 10к; R52, R54 - 510Ом;
25	DA18	C16	R7, R10, R13, R16 - R19, R22, R25, R28 - 200Ом;
26	DA19	B28	Резисторы ЧП: R51 - 200 Ом;
27	DA20	C15	R56 - 4,7к; R60 - 2к; C1, C2 - 47 пФ;
28	DA21	B27	C4, C5 - 300 пФ; C3, C6... C29 - 0,1 мФ.
29	DA22	C14	
30	DA23	B26	
31	DA24	C13	
32	DA25	B25	
33	DA26	C12	
34	DA27	B24	
35	DA28	C11	
36	DA29	B23	
37	DA30	C10	
38	DA31	B22	
39	DA32	C9	
40	DA33	B21	
41	DA34	C8	
42	DA35	B20	
43	DA36	C7	
44	DA37	B19	
45	DA38	C6	
46	DA39	B18	
47	DA40	C5	
48	DA41	B17	
49	DA42	C4	
50	DA43	B16	
51	DA44	C3	
52	DA45	B15	
53	DA46	C2	
54	DA47	B14	
55	DA48	C1	
56	DA49	B13	
57	DA50	C0	
58	DA51	B12	
59	DA52	C-1	
60	DA53	B11	
61	DA54	C-2	
62	DA55	B10	
63	DA56	C-3	
64	DA57	B9	
65	DA58	C-4	
66	DA59	B8	
67	DA60	C-5	
68	DA61	B7	
69	DA62	C-6	
70	DA63	B6	
71	DA64	C-7	
72	DA65	B5	
73	DA66	C-8	
74	DA67	B4	
75	DA68	C-9	
76	DA69	B3	
77	DA70	C-10	
78	DA71	B2	
79	DA72	C-11	
80	DA73	B1	
81	DA74	C-12	
82	DA75	B0	
83	DA76	C-13	
84	DA77	B-1	
85	DA78	C-14	
86	DA79	B-2	
87	DA80	C-15	
88	DA81	B-3	
89	DA82	C-16	
90	DA83	B-4	
91	DA84	C-17	
92	DA85	B-5	
93	DA86	C-18	
94	DA87	B-6	
95	DA88	C-19	
96	DA89	B-7	
97	DA90	C-20	
98	DA91	B-8	
99	DA92	C-21	
100	DA93	B-9	

XP2

A	АЛЛ	В30
B	ПСЕН	С7
	РЕСЕТ	Е31
1	Т0	С3
2	Т1	В3
3	Т2	С5
4	Т3	В5
5	Т4	С6
6	Т5	В6
7	Т6	С7
8	Т7	В7
9	РА0	СВ
10	РА1	ВВ
11	РА2	С9
12	РА3	В9
13	РА4	С10
14	РА5	В10
15	РА6	С11
16	РА7	В11
17	РА8	С12
18	РА9	В12
19	РА10	С13
20	РА11	В13
21	РА12	С14
22	РА13	В14
23	РА14	С15
24	РА15	В15
25	РА16	С16
26	РА17	В16
27	РА18	С17
28	РА19	В17
29	РА20	С18
30	РА21	В18
31	РА22	С19
32	РА23	В19
33	РА24	С20
34	РА25	В20
35	РА26	С21
36	РА27	В21
37	РА28	С22
38	РА29	В22
39	РА30	С23
40	РА31	В23

Статья поступила 14.12.87

А. О. Воробьев, А. Б. Слюсар

ПРОГРАММА ВЫЧИСЛЕНИЯ КВАДРАТНОГО КОРНЯ ДЛЯ ОЭВМ СЕРИИ К1816

Для некоторого класса задач, например статистической обработки данных, адаптивной цифровой фильтрации и других, возникает необходимость вычисления квадратного корня $Y=X$ из многобайтного целого числа. Предлагаемая подпрограмма SQRT вычисляет корень из 3- или 2-байт целого положительного числа со знаковым старшим разрядом. Подкорневое число размерностью NBIN байт располагается в области внутреннего ОЗУ данных с начального адреса X. Результат работы подпрограммы (2-байт число) размещается с начального адреса Y. Указатели X и Y адресуют младшие байты числа.

В основе алгоритма подпрограммы лежит итерационная формула

$$Y_{i+1} = (X/Y_i + Y_i)/2,$$

где X — подкорневое число; Y_i — приближенное значение корня. Число итераций, необходимых для вычисления корня, сильно зависит от выбора начального приближения Y и может быть довольно большим. Для уменьшения числа итераций приближение Y следует выбирать по возможности близким к истинному значению корня. В программе Y устанавливается равным целому значению корня из веса старшей значащей единицы подкорневого числа. Все начальные приближения сведены в таблицу. Исследования показали, что такой выбор начального приближения позволяет сократить число итераций, необходимых для получения значения корня с точностью до целой части, до трех.

Время вычисления корня по приведенной формуле в значительной степени зависит от скорости выполнения операции деления, которая реализуется подпрограммой DIV. Заметим, что деление должно выполняться с 3- или 2-байт делимым (подкорневое число X) и 2-байт делителем (приближение результата Y_i). В подпрограмме DIV применен быстрый алгоритм деления без восстановления остатка с одновременным сдвигом делимого и результата. Восстановление остатка можно исключить на j-м шаге, если на следующем (j+1)-м шаге вместо вычитания делителя. Та-

LOC	OBJ	LINE	SOURCE STATEMENT
		1	*****
		2	* SQRT ПОДПРОГРАММА ВЫЧИСЛЕНИЯ КВАДРАТНОГО КОРНЯ *
		3	*****
		4	;
0020		5	X EQU 20H ; 3 БАЙТА ПОДКОРНЕВОГО ЧИСЛА
0023		6	YI EQU 23H ; 2 БАЙТА ПРИБЛИЖЕН. ЗНАЧ. КОРНЯ Y(I)
0025		7	YY EQU 25H ; 2 БАЙТА ПРИВЕДЕННОГО ЗНАЧ. Y(I)
0027		8	YVC EQU 27H ; 2 БАЙТА ПРИВЕДЕННОГО ЗНАЧ. -Y(I)
0003		9	NBIN EQU 3 ; РАЗМЕРНОСТЬ ПОДКОРНЕВОГО ЧИСЛА
		10	;
019A		11	LOADY EQU LOAD2
01A6		12	SHFTY EQU SHFT2
		13	;
		14	IF NBIN EQ 3
0197		15	LOADX EQU LOAD3
01A3		16	SHFTX EQU SHFT3
01A0		17	SHFTD EQU SHFT4
000A		18	CR EQU 10
		19	;
		20	ELSE
		21	LOADX EQU LOAD2
		22	SHFTX EQU SHFT2
		23	SHFTD EQU SHFT3
		24	CR EQU 2
		25	ENDIF
		26	;
0100		27	ORG 100H
		28	;
0100	B920	29	SQRT: MOV R1, X ; ПРОВЕРКА ПОДКОРНЕВОГО ЧИСЛА НА
0102	BA03	30	MOV R2, NBIN; РАВЕНСТВО 0
0104	F1	31	MOV A, R1
0105	960E	32	JNZ S1
0107	19	33	INC R1
0108	EA04	34	DJNZ R2, S0
010A	A1	35	MOV @R1, A ; РЕЗУЛЬТАТ РАВЕН 0
010B	19	36	INC R1
010C	A1	37	MOV @R1, A
010D	93	38	RETR ; ВЫХОД
		39	;
010E	343E	40	S1: CALL SQX ; ВЫЗОВ П/П ПРИВЕДЕНИЯ ПОДКОРНЕВОГО ЧИСЛА
0110	2316	41	MOV A, #22 ; ВЫЧИСЛЕНИЕ СМЕЩЕНИЯ ДЛЯ ТАБЛИЦЫ
0112	37	42	CPL A ; ПРИБЛИЖЕННЫЙ Y0
0113	6A	43	ADD A, R2
0114	37	44	CPL A
0115	E7	45	RL A
0116	03AD	46	ADD A, #LOW TABY0
0118	A8	47	MOV R0, A
0119	B923	48	MOV R1, #YI ; ЗАГРУЗИТЬ ПО АДР. YI ЗНАЧЕНИЕ
011B	A3	49	MOVP A, #0 ; НУЛЕВОГО ПРИБЛИЖЕНИЯ
011C	A1	50	MOV @R1, A
011D	18	51	INC R0
011E	19	52	INC R1
011F	F8	53	MOV A, R0
0120	A3	54	MOVP A, #0
0121	A1	55	MOV @R1, A
0122	B803	56	MOV R0, #3 ; В R0 СЧЕТЧИК ИТЕРАЦИЙ
		57	;
0124	345C	58	ITR: CALL SQY ; ВЫЗОВ П/П ПРИВЕДЕНИЯ ДЛЯ Y
0126	B920	59	MOV R1, X ; ЗАГРУЗКА В R7, R6, R5 ПОДКОРНЕВОГО ЧИСЛА
		60	;
0128	3497	61	CALL LOADX
012A	347D	62	CALL DIV ; ВЫЗОВ П/П ДЕЛЕНИЯ
012C	B923	63	MOV R1, #Y1
012E	FC	64	MOV A, R4
012F	61	65	ADD A, @R1 ; СЛОЖЕНИЕ ЧАСТНОГО С ПРИБЛИЖЕНИЕМ Y1
0130	AC	66	MOV R4, A
0131	19	67	INC R1
0132	FD	68	MOV A, R5
0133	71	69	ADDC A, @R1
0134	97	70	CLR C
0135	67	71	RRC A ; ДЕЛЕНИЕ СУММЫ НА 2
0136	A1	72	MOV @R1, A ; ЗАПИСЬ РЕЗУЛЬТАТА ПО АДРЕСУ Y1
0137	C9	73	DEC R1
0138	FC	74	MOV A, R4
0139	67	75	RRC A
013A	A1	76	MOV @R1, A
013B	E824	77	DJNZ R0, ITR ; ЕСЛИ R0=0, ВЫПОЛНИТЬ НОВУЮ ИТЕРАЦИЮ
013D	93	78	RETR ; ВЫХОД ПО АДР. Y1 ЗНАЧЕНИЯ КОРНЯ
		79	;
		80	SQX: ; П/П ПРИВЕДЕНИЯ ПОДКОРНЕВОГО ЧИСЛА (3 БАЙТА ПО АДР. X)
		81	; РЕГ R2 - СЧЕТЧИК СДВИГОВ ПРИВЕДЕНИЯ
013E	97	82	CLR C
013F	B920	83	MOV R1, X ; ЗАГРУЗИТЬ 3 БАЙТА ПОДКОРНЕВОГО ЧИСЛА В РЕГ. R7, R6, R5
0141	3497	84	CALL LOADX
0143	BA00	85	MOV R2, #0
0145	D24C	86	MO: JB6 M1 ; СДВИГ ВЛЕВО ДО ПОЯВЛЕНИЯ 1 В РАЗРЯДЕ 6 И ПОДСЧЕТ ЧИСЛА СДВИГОВ
0147	34A3	87	CALL SHFTX

LOC	OBJ	LINE	SOURCE STATEMENT
0149	1A	88	INC R2 ; В R2
014A	2445	89	JMP MO
014C	B920	90 M1:	MOV R1, #X ; ЗАПИСЬ РЕЗУЛЬТАТА ПО АДР. X
014E	FD	91	MOV A, R5
014F	A1	92	MOV @R1, A
0150	19	93	INC R1
0151	FE	94	MOV A, R6
0152	A1	95	MOV @R1, A
0153	19	96	INC R1
0154	FF	97	MOV A, R7
0155	A1	98	MOV @R1, A
0156	FA	99	MOV A, R2 ; КОРРЕКЦИЯ ЧИСЛА СДВИГОВ И
0157	37	100	CPL A ; СОХРАНЕНИЕ ЕГО В R3
0158	030A	101	ADD A, #CR
015A	AB	102	MOV R3, A
015B	83	103	RET
		104 ;	
		105 SQY:	; П/П ПРИВЕДЕНИЯ Y1
		106	; ПРИВЕДЕННОЕ ЗНАЧЕНИЕ Y1 ЗАНОСИТСЯ
		107	; ПО АДР. Y1 - (ПОЛОЖИТ. ЧИСЛО)
		108	; И ПО АДР. Y0C - (ОТРИЦАТ. ЧИСЛО)
015C	97	109	CLR C
015D	FB	110	MOV A, R3 ; В R2 СКОРРЕКТИРОВАННОЕ ЧИСЛО СДВИГОВ
015E	AA	111	MOV R2, A ; ПРИВЕДЕНИЯ ПОДКОРНЕВОГО ЧИСЛА
015F	B923	112	MOV R1, #Y1 ; ЗАГРУЗКА 2-БАЙТНОГО КОДА ПРИБЛИЖЕНИЯ
0161	349A	113	LOADY ; В РЕГ. R7, R6
0163	D26A	114 M2:	JB6 M3 ; СДВИГ ВЛЕВО ДО ПОЯВЛЕНИЯ 1 В РАЗРЯДЕ 6
0165	34A6	115	CALL SHFTY ; И ПОДСЧЕТ ЧИСЛА СДВИГОВ В R2
0167	1A	116	INC R2
0168	2463	117	JMP M2
016A	B926	118 M3:	MOV R1, #YY+1 ; ЗАПИСЬ РЕЗУЛЬТАТА
016C	A1	119	MOV @R1, A
016D	C9	120	DEC R1
016E	FE	121	MOV A, R6
016F	A1	122	MOV @R1, A
0170	B927	123	MOV R1, #YYC
0172	37	124	CPL A
0173	0301	125	ADD A, #1
0175	A1	126	MOV @R1, A
0176	19	127	INC R1
0177	FF	128	MOV A, R7
0178	37	129	CPL A
0179	1300	130	ADDC A, #0
017B	A1	131	MOV @R1, A
017C	83	132	RET
		133 ;	
		134 DIV:	; П/П ДЕЛЕНИЯ. ПРИВЕДЕННОЕ ДЕЛИМОЕ В РЕГ. R7, R6, R5,
		135	; ПРИВЕДЕННЫЙ ДЕЛИТЕЛЬ ПО АДР. Y1 (ПОЛОЖИТ. ЧИСЛО)
		136	; И Y0C (ОТРИЦАТ. ЧИСЛО) В РЕГ. R2 ЧИСЛО ЦИКЛОВ ДЕЛЕНИЯ
		137	; РЕЗУЛЬТАТ В РЕГ. R5, R4
017D	27	138	CLR A
017E	AC	139	MOV R4, A
017F	85	140	CLR F0
0180	95	141	CPL F0 ; F0 = 1
0181	FE	142 DIV0:	MOV A, R6
0182	B927	143	MOV R1, #YYC
0184	B668	144	JFO DIV1
0186	B925	145	MOV R1, #YY
0188	61	146 DIV1:	ADD A, @R1 ; ВЫЧИТАНИЕ ДЕЛИТЕЛЯ, ЕСЛИ F0=1,
0189	AE	147	MOV R6, A ; И СЛОЖЕНИЕ, ЕСЛИ F0=0
018A	FF	148	MOV A, R7
018B	19	149	INC R1
018C	71	150	ADDC A, @R1
018D	AF	151	MOV R7, A
018E	85	152	CLR F0
018F	E692	153	JNC DIV2 ; ЕСЛИ ПЕРЕНОС C=1, ТО РАЗРЯД ЧАСТНОГО
0191	95	154	CPL F0 ; РАВЕН 1 И F0=1
0192	34A0	155 DIV2:	CALL SHFTD
0194	EAB1	156	DJNZ R2, DIV0 ; ЦИКЛ ПО R2
0196	83	157	RET
		158 ;	
		159 ;	П/П ЗАГРУЗКИ РАБОЧИХ РЕГИСТРОВ
		160 ;	
0197	F1	161 LOAD3:	MOV A, @R1
0198	AD	162	MOV R5, A
0199	19	163	INC R1
019A	F1	164 LOAD2:	MOV A, @R1
019B	AE	165	MOV R6, A
019C	19	166	INC R1
019D	F1	167	MOV A, @R1
019E	AF	168	MOV R7, A
019F	83	169	RET
		170 ;	
		171 ;	П/П СДВИГА ВЛЕВО НА 1 РАЗРЯД СОДЕРЖИМОГО РАБОЧИХ РЕГИСТРОВ
		172 ;	
01A0	FC	173 SHFT4:	MOV A, R4
01A1	F7	174	RLC A
01A2	AC	175	MOV R4, A
01A3	FD	176 SHFT3:	MOV A, R5
01A4	F7	177	RLC A

ким образом, на каждом шаге необходимо либо вычесть, либо прибавить делитель, в зависимости от знака остатка. В системе команд ОЭВМ серии K1816* отсутствует команда вычитания. Вместо нее следует использовать сложение дополнительных кодов. Число шагов деления подпрограммы DIV переменное и устанавливается в зависимости от количества значащих разрядов делимого и делителя. Так, например, если делимое имеет 17 значащих разрядов, а делитель — 11, то необходимо выполнить $17 - 11 + 1 = 7$ шагов деления для получения результата с точностью до целой части.

Для определения числа значащих разрядов операндов и числа необходимых шагов деления перед вызовом подпрограммы DIV работают две подпрограммы SQX и SQY, в которых выполняется приведение (сдвиг влево до появления единицы в старшем после знакового разряде) делимого и делителя. Подпрограмма SQX вызывается один раз, так как делимое (аргумент X в формуле) остается постоянным на всех итерациях. Подпрограмма SQY вызывается на каждой итерации для приведения нового Y1. Приведенное значение приближения Y1 и его дополнительный код запоминаются для использования в подпрограмме DIV. Суммарное число сдвигов приведения подпрограмм SQX и SQY определяет число шагов деления. Кроме того, число сдвигов подпрограммы SQX используется для образования смещения при обращении к таблице TABYO для извлечения начального приближения Y. Все операции сложения выполняются, как с целыми 2-байт числами со знаком. Деление на два осуществляется сдвигом вправо на один разряд.

Текст подпрограммы SQRT с комментариями приводится для случая расположения ее, начиная с адреса 100H. Объем подпрограммы SQRT составляет 171 байт. Таблица начальных приближений занимает 46 байт. Ее следует располагать на одной странице памяти команд с подпрограммой SQRT. Возможна замена команды MOVP A, @A в строках 49 и 54 на команду MOVP3 A, @A для размещения таблицы TABYO в странице 3 памяти команд. Время вычисления квадратного корня зависит от абсолютной величины аргумента X и не превышает 3,5 мс (тактовая частота генератора 6 МГц).

* Кобылинский А. В., Линовецкий Г. П. Однокристальные микроЭВМ серии K1816//Микропроцессорные средства и системы, — 1986. — № 1, — С. 10—19.

LOC	OBJ	LINE	SOURCE	STATEMENT	Окончание программы
01A5	AD	178	MOV	R5,A	
01A6	FE	179	SHFT2:	MOV	A,R6
01A7	F7	180	RLC	A	
01A8	AE	181	MOV	R6,A	
01A9	FF	182	MOV	A,R7	
01AA	F7	183	RLC	A	
01AB	AF	184	MOV	R7,A	
01AC	B3	185	RET		
		186 ;			
		187	TABYO:		; ТАБЛИЦА НУЛЕВЫХ ПРИБЛИЖЕНИЙ Y0
01AD	O1	188	DB	1,0	; /1
01AF	O1	189	DB	1,0	; /2
01B1	O2	190	DB	2,0	; /4
01B3	O2	191	DB	2,0	; /8
01B5	O4	192	DB	4,0	; /16
01B7	O5	193	DB	5,0	; /32
01B9	O8	194	DB	8,0	; /64
01BB	OB	195	DB	11,0	; /128
01BD	10	196	DB	16,0	; /256
01BF	16	197	DB	22,0	; /512
01C1	20	198	DB	32,0	; /1024
01C3	2D	199	DB	45,0	; /2048
01C5	40	200	DB	64,0	; /4096
01C7	5A	201	DB	90,0	; /8192
01C9	80	202	DB	128,0	; /16384
		203	IF	NBIN	EQ 3
01CB	B5	204	DB	181,0	; /32768
01CD	O0	205	DB	0,01	; /65536
01CF	6A	206	DB	6AH,01	; /131072
01D1	O0	207	DB	0,02	; /262144
01D3	D4	208	DB	0D4H,02	; /524288
01D5	O0	209	DB	00,04	; /1048576
01D7	AB	210	DB	0ABH,05	; /2097152
01D9	FF	211	DB	0FFH,07	; /4191304
		212	ENDIF		
		213 ;			
		214	END		

USER SYMBOLS

CR	000A	DIV	017D	DIV0	0181	DIV1	0188	DIV2	0192	ITR	0124
LOAD2	019A	LOAD3	0197	LOADX	0197	LOADY	019A	MO	0145	M1	014C
M2	0163	M3	016A	NBIN	0003	SO	0104	S1	010E	SHFT2	01A6
SHFT3	01A3	SHFT4	01A0	SHFTD	01A0	SHFTX	01A3	SHFTY	01A6	SQRT	0100
SQX	013E	SQY	015C	TABYO	01AD	X	0020	YI	0023	YY	0025
YXC	0027										

ASSEMBLY COMPLETE, NO ERRORS

Телефон: 229-24-07, Киев, Андрей Борисович Слюсар

Статья поступила 25.08.88

ИНФОРМАЦИЯ

Московский экспериментальный вычислительный центр «Элекс» ГКВТИ СССР и компания «Вест интернешнл» (Норвегия) подписали учредительные документы по созданию совместного советско-норвежского предприятия «Элекс-Вест».

С советской стороны — заместитель председателя ГКВТИ СССР В. В. Корчагин и директор МЭВЦ «Элекс» И. З. Карась, с норвежской — председатель фирмы Э. Мустад и президент Л. Халворсен.

Задачи совместного предприятия — производство персональных ЭВМ, разработка, сопровождение и экспорт программных средств, создание автоматизированных рабочих программно-технических комплексов и систем.

Были также подписаны протоколы о сотрудничестве в области информатики между Кооперативным Союзом «Информатика» (СССР) и фирмами «Вест интернешнл» (Норвегия), «Эплайд майкро электроникс лимитид» (Ирландия) и «Трейс продакес» (США).

Кооперативный Союз «Информатика» и совместное советско-норвежское предприятие «Элекс-Вест» объявили конкурс среди программистов на лучшую разработку игровой и обучающей программ для персональных ЭВМ, производимых на западном рынке. Победителю конкурса присуждается приз — персональная ЭВМ фирмы «Элекс-Вест» (класса PC/AT) или премия в размере 50 тысяч рублей (по выбору победителя).

Пресс-бюро МЭВЦ «Элекс», тел. 208-66-18, Москва

УДК 681.32+681.3.06

И. Г. Муттик, Н. А. Самарин, В. Ковачик

СВЯЗЬ ПЭВМ ЕС1841 С МИКРОЭВМ «ЭЛЕКТРОНИКА ДЗ-28»

Разработан аппаратно-программный комплекс, обеспечивающий совместную работу ПЭВМ ЕС1841 и микроЭВМ ДЗ-28. При создании такого комплекса за основу были приняты минимизация аппаратных затрат и перенос основных функций на программное обеспечение. Выбран протокол межмашинного обмена по двум токовым петлям (ТП) в одну сторону (с максимальным удалением машин друг от друга на 100 м) без подтверждения приема, с передачей контрольных битов четности. Одна ТП используется для передачи информации, а другая — для синхронизации ее приема в ПЭВМ. Числовые данные и программы передаются по линии связи как последовательность битов (группы по 9 бит). Каждый бит в информационной ТП сопровождается импульсом в ТП синхронизации. Узел, осуществляющий передачу из ДЗ-28, состоит из двух ИМС: К155ЛА3 и К155ЛА8; принимающий узел — из двух оптронов АОТ110 и БИС К580ВВ55, сопряженной с каналом ПЭВМ (она занимает в пространстве портов адреса 71С ... 71Е НЕХ).

Программы, осуществляющие протокол пересылки, написаны в кодах обеих машин. Программа в ДЗ-28 обеспечивает выдачу в линию связи байтов и чисел (из регистра X). Программа в ПЭВМ осуществляет прием и контролирует неискаженность данных. Скорость передачи примерно равна 500 бит/с.

На языке БЕЙСИК разработаны кроссассемблер и дисассемблер для ДЗ-28. Текст для обработки ассемблером может быть подготовлен с помощью любого текстового процессора. Ассемблер позволяет передавать результат ассемблирования в файл на диске, на принтер, в линию связи с ДЗ-28, а также формировать листинг программы с машинными кодами ДЗ-28. Сообщения об ошибках «привязаны» к строкам и поэтому позволяют быстро найти неточности и быстро исправить.

Дисассемблер обрабатывает файлы, содержащие программу для ДЗ-28. В результате дисассемблирования создается текстовый файл, содержащий мнемоники машинных кодов ДЗ-28, а также сами машинные коды в виде гексадцимальных цифр в поле комментария к каждой строке.

Комплекс, состоящий из ПЭВМ ЕС1841 и двух ДЗ-28, успешно эксплуатируется на Физическом факультете МГУ в течение года. Программные компоненты комплекса хорошо протестированы.

Телефон 939-11-47, Москва, Физический факультет МГУ
Сообщение поступило 06.07.87

УДК 681.3.06

В. М. Вуколиков, Ю. Н. Дроздов, В. В. Колобков,
А. В. Лазарев, Е. П. Панкрац

ПАКЕТ ПРОГРАММ УПРАВЛЕНИЯ ПРИБОРАМИ ЧЕРЕЗ МАГИСТРАЛЬ МЭК 625.1 ДЛЯ ПЭВМ

Приборный интерфейс (ПИ) или интерфейс МЭК целесообразно использовать для построения простых и дешевых систем, в состав которых может входить до 31 прибора при максимальной протяженности магистрали до 20 м и скоростях обмена данными до 1 Мбайт/с.

Пакет программ для управления аппаратурой на магистрали ПИ от ПЭВМ типа IBM PC реализован в операционной системе MS DOS на языке ассемблера с системой команд Intel 8086 и адаптирован для работы на языке Фортран 77 (рис. 1). Для организации взаимодействия с интерфейсной аппаратурой на магистрали МЭК 625.1 пользователь составляет Фортран-программу, с помощью которой обращается к программам пакета, используя операторы CALL.

Все программы можно условно разделить на четыре группы.

Программы группы 1 (табл. 1) выполняют операции, в ходе которых плата сопряжения (ПС) находится в состоянии «источник» или «приемник». К ним относятся операции блочного обмена между ОЗУ ПЭВМ (файлом на диске) и магистралью ПИ, а также операция перевода ПС в состояние «контроллер» (RECCNT).

Группу 2 (табл. 2) составляют программы, выполняющие операции, в ходе которых ПС должна находиться в состоянии активного контроллера. Особую роль играет подпрограмма INIT (инициализация). Она не только проводит общий сброс магистрали, но и готовит к работе остальные программы пакета — задает базовые адреса, режим обработки прерываний и определяет программу завершения задачи.

Программы группы 3 (табл. 3) выполняют операции настройки ПС на работу в режиме системного контроллера магистрали: аварийный сброс интерфейса, перевод устройств (приборов) в дистанционный и местный режимы управления.

Программы группы 4 (табл. 4) реализуют операции-примитивы, используемые при исполнении программ первых трех групп. Они работают, когда ПС является активным контроллером: назначают и программируют

функции приборов, а также выполняют базовые операции управления магистралью ПИ.

При обращении к подпрограмме на ассемблере используется вызов типа FAR [1, 2] с передачей четырех байтов на каждый параметр, где два первых байта содержат адрес сегмента, а 3-й и 4-й байты — относительное смещение в этом сегменте. Таким образом, чтобы получить истинное значение параметра, нужно прочитать значение, расположенное в указанном сегменте и смещенное от его начала. Такая информация о каждом передаваемом параметре (независимо от его типа) заносится в стек, начиная с первого. Последним заносится 4-байтовый адрес возврата подпрограммы. Каждая вызываемая процедура должна включать число передаваемых параметров.

При разработке ПО использовались четыре резидентные программы, приспособленные для вызова с языка БЕЙСИК и обмена между магистралью ПИ и памятью. Адрес каждого параметра представлен двумя байтами, причем порядок передачи параметров обратный по отношению к Фортрану. Кроме того, для обращения к нужной программе необходимо косвенно задать 4-байтовый адрес вызываемой процедуры, точно указывая адрес сегмента и величину смещения.

Таблица 1

Программы, не выполняющие функции контроллера

Наименование	Параметры	Выполняемые функции
READST	TA, N, EOS, TERM, M, STRING	Чтение строки данных STRING длиной M (STRING, M — выходные параметры)
WRITST	LSA, N, EOS, TERM, STRING, M	Запись строки данных STRING длиной M (все параметры входные)
RDSTRD	TA, N, EOS, TERM, STRING, M	Чтение строки данных по каналу прямого доступа (STRING, M — выходные параметры)
WRSTRD	LSA, N, EOS, TERM, STRING, M	Запись строки данных по каналу прямого доступа (все параметры входные)
RDTOFD	TA, N, EOS, TERM, STR1	Чтение данных в файл по каналу прямого доступа (все параметры входные)
WRFRFD	LSA, N, EOS, TERM, STR1	Запись данных из файла по прямому доступу (все параметры входные)
RECCNT	Нет параметров	Получить управление магистралью ПИ

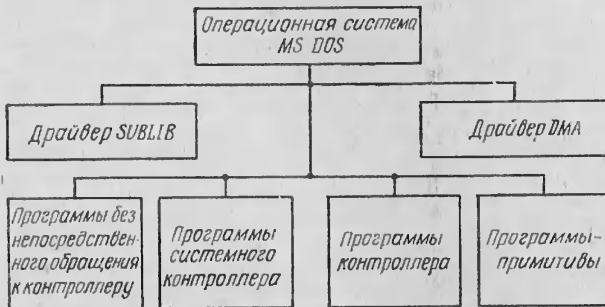


Рис. 1. Общая структура пакета программ

Таблица 2

Программы, реализующие функции контроллера

Наименование	Параметры	Выполняемые функции
ADC	Нет параметров	Универсальный сброс всех устройств
GET	LA, N	Запуск групповой операции (параметры входные)
SDL	LA, N	Перевод указанных устройств в локальный режим (параметры входные)
INIT	MYADDR, BDADDR, TCIMODE, MSV	Инициализация (все параметры входные)
LLO	Нет параметров	Местная блокировка
PARPOL	ROLRESP	Параллельный опрос (параметр выходной)
PPD	LA, N	Запрет параллельного опроса указанных устройств (параметры входные)
PPU	Нет параметров	Отмена параллельного опроса
RBST	STATUS	Чтение состояния шины ПИ (параметр входной)
PASCNT	TDA	Передача управления (параметр входной)
SDC	LA, N	Сброс указанных устройств (параметры входные)
SERPOL	TDA, POLRESP	Последовательный опрос (POLRESP — выходной параметр)
PPEN	LDA, BIT SENSE	Разрешение параллельного опроса (все параметры входные)
TRANSF	LSA, N, TA, M, EOS, TERM	Передача данных от источника к приемникам (все параметры входные)

Таблица 3

Программы команд системного контроллера

Наименование	Параметры	Выполняемые функции
ABORT	Нет параметров	Аварийный сброс интерфейса
ADTR	Нет параметров	Перевод всех устройств в дистанционный режим управления
ADTL	Нет параметров	Перевод всех устройств в местный режим управления
SDL	LA, N	Перевод указанных устройств в местный режим управления (все параметры программ входные)

Примечание к таблицам. Описание и список параметров

STRING, STR1 — символьные (байтовые) параметры.

LAST — логический параметр.

Параметры подпрограмм:

- LA — список приборов-приемников (массив).
- LSA — список приборов-приемников и вторичные адреса (массив).
- LDA — адреса прибора-приемника.
- TA — прибор-источник и вторичные адреса (массив).
- TDA — адрес прибора-источника.
- N, M — длина массива.
- EOS — ASCII-код символа, завершающего обмен.
- TERM — тип завершения обмена: по линии EOI или EOS.
- MYADDR — адрес платы интерфейса КОП на шине ПИ.
- BDADDR — базовый адрес контроллера КОП в пространстве ввода-вывода.
- TCIMODE — режим прерывания по завершению задачи на магистрали КОП.
- MSV — дополнительная характеристика режима завершения задачи.
- BIT — номер бита в байте.
- SENSE — значение, выставляемое битом: 0 или 1.
- POLRESP — ответ на параллельный (последовательный) опрос устройств.
- STATUS — статус шины ПИ.
- LAST — признак последнего байта обмена (логический).
- SYMBOL — ASCII-код символа.
- STRING — строка прочитанных (записанных) байтов (символьный массив).
- STR1 — имя файла в ASCII-кодах (символьный массив).

Для выдачи сообщений из внутренних подпрограмм на ассемблере, например при неправильном обращении к программе или ошибочной ситуации, рекомендуется [1] пользоваться директивной печати текстовой строки операционной системы MS DOS (прерывание INT 21H, запрос 9). Однако особенности реализации транслятора с языка Фортран сделали этот способ неприемлемым. Поэтому при выводе диагностических сообщений не использовался запрос операционной системы MS DOS (прерывание по вектору 10H), который осуществляет телетайпный вывод текстовой информации через видеопамять на русском языке.

Для сопряжения ПЭВМ с магистралью ПИ применялась ПС фирмы Scientific Solutions IEEE 488 Interface [3] в составе блоков приемника-источника, контроллера, сопряжения (рис. 2). Блок приемник-источник, реализованный на БИС фирмы Intel 8291A, выполняет стандартные функции на магистрали ПИ, за исключением функций контроллера. Блок контроллера в виде БИС 18291 работает как ведомый процессор с ведущей ЭВМ

Таблица 4

Программы, реализующие команды-примитивы		
Наименование	Параметры	Выполняемые функции
GTSB	Нет параметров	Переключить интерфейс в резерв
MLA	Нет параметров	Собственный адрес контроллера как приемника на магистрали КОП
MTA	Нет параметров	Собственный адрес контроллера как источника на магистрали КОП
SETEOI	Нет параметров	Установить линию EOI
TASY	Нет параметров	Захватить управление асинхронно
TCSY	Нет параметров	Захватить управление асинхронно
UNTLIST	Нет параметров	Установить сигнал UNL (не принимать)
UNTALK	Нет параметров	Выставить сигнал UNT (не передавать)
TALK	TA, N	Назначить устройство источником (параметры программы входные)
LISTEN	LSA, N	Назначить устройства-приемники (параметры программы входные)
RDBYTE	SYMBOL, LAST	Прочитать байт SYMBOL (параметры программы выходные)
WRBYTE	SYMBOL, LAST	Записать байт SYMBOL (параметры программы выходные)
GTSBCT	Нет параметров	Переключить интерфейс в резерв, не сбрасывая счетчик
GIDL	Нет параметров	Освободить интерфейс

и выполняет функции контроллера ПИ. Электрическое сопряжение между ПС и магистралью осуществляется с помощью двух блоков сопряжения (БИС 18293). Эти двунаправленные формирователи специально разработаны для применений на интерфейсной магистрали ПИ.

Для передачи данных могут использоваться операции прямого доступа в память. Информация о состоянии 18291А может передаваться по прерыванию или по опросу. Схема 18292 выполняет контроллерные функции. Связь с ведущим процессором осуществляется по прерыванию или по опросу.

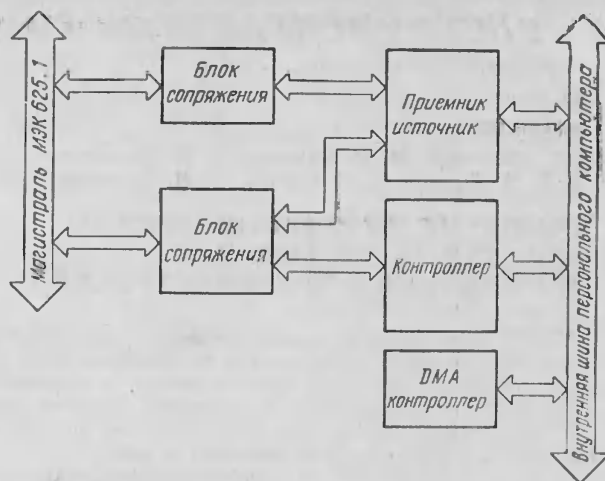


Рис. 2. Структурная схема платы сопряжения приборного интерфейса

Аналогом БИС 18291А является микросхема КР580ВК91, БИС 18292 — КР580ВК92. Электрическое сопряжение с магистралью ПИ может быть выполнено на микросхемах КР580ВВ51 и К1801ВП-035 (18293) [4].

Совместимость отечественных БИС, используемых для управления ПИ, с БИС, которые применены в ПС, позволяет рассчитывать на возможность сравнительно простой адаптации разработанного пакета подпрограмм к отечественным средствам сопряжения ПЭВМ с ПИ.

При отладке программного обеспечения для работы с ПИ использовался пульт-имитатор (индикатор) сигналов в МЭКе, собранный на микросхемах серии К155 в конструктиве модуля КАМАК. Индикатор позволил заменить промышленные логические анализаторы ЛКП 814 и ЛКП 820, выпускаемые для аналогичных целей.

Схема индикатора состоит из 16 каскадов четырех типов:

тип 1 (один каскад) обеспечивает визуальную индикацию и имитацию сигнала IFC;

тип 2 (два каскада) индицирует и имитирует сигналы NRFD и NDAC;

тип 3 (11 каскадов) обеспечивает визуальную индикацию задания сигналов данных ПИ D10—D107, а также позволяет индицировать сигналы EOI, DAV и SRQ;

тип 4 (два каскада) позволяет индицировать сигналы ATN и REN.

Скорость передачи данных от прибора-источника к прибору-приемнику при программировании с помощью нашего пакета составляет 20...25 Кбайт/с. Пакет в объектном коде занимает около 11 Кбайт. Он снабжен руководством программиста, содержащим достаточно полную инструкцию по применению.

Телефон 466-20-13, Москва

ЛИТЕРАТУРА

1. MS DOS Technical Reference.— MicroSoft Corp., 1985.
2. MS DOS FORTRAN Programmer's Guide MicroSoft Corp., 1985.
3. Scientific Solutions IEEE 488 Interface.— Scientific Solutions Incorporated, 1985. Vol. 18.— N 8.— P. 442—444.
4. Мячев А. А., Снегирев А. А. Контроллеры приборных интерфейсов мини- и микроЭВМ // Микропроцессорные средства и системы.— 1984.— № 4.— С. 32—39.

Статья поступила 03.07.87

УДК 681.325.5—181.4

Г. Г. Мамджян, М. П. Наумчик, А. Н. Прохоренко, Б. Я. Розман, Д. С. Горбов, Н. И. Рузанова

КОМПЛЕКС МИКРОПРОЦЕССОРНЫХ МОДУЛЬНЫХ СРЕДСТВ ДЛЯ ПРИМЕНЕНИЯ В ПРИБОРОСТРОЕНИИ

ВНИИ медицинского приборостроения разработал и выпускает комплекс аппаратных и программных средств, обеспечивающих процессы проектирования и производства электронных приборов и аппаратов (прежде всего, медицинского назначения) с встроенными микропроцессорами (МП). Комплекс включает в себя:

- 8- и 16-битные наборы встраиваемых унифицированных МП модулей (ТЭЗов);
- средства проектирования, отладки и тестирования;
- системное, прикладное, технологическое ПО;
- унифицированные конструкции.

Наборы встраиваемых МП модулей обеспечивают решение на современном научно-техническом уровне типовых задач приборостроения (обслуживание; клавиатуры; индикация и визуализация информации; диалог с оператором; съем, контроль, предварительная обработка, обработка информации; управление; самопроверка, самотестирование, сигнализация; связь с центральной ЭВМ и другими приборами).

Комплекс включает в себя четыре набора модулей:

№ набора	Тип главного процессора (серия)	Разрядность, бит	Количество модулей в наборе (номенклатура)	Конструктив*	Системный интерфейс	Периферийный интерфейс	Средства отображения информации
1	K1816	8	3	E1.1	—	Специальный последовательный канал	Символьные индикаторы
2	K580	8	12	E1.2	И41М	ИРПС, ИРПР	ЭЛТ; бегущая строка
3	K580	8	3	E2.1 E2.2	И41М	ИРПС Параллельный	Бегущая строка; ЭЛТ
4	K1810	16	3	E2.2	И41М	ИРПР, ИРПС	Бегущая строка; ЭЛТ

* E1.1—100×160 мм, E1.2—100×220 мм, E2.1—233×160 мм, E2.2—233×220 мм

В зависимости от сложности решаемых задач и стоимости аппаратуры заказчик может выбрать из предлагаемой номенклатуры наиболее оптимальную конфигурацию модулей.

Предлагаемые технические средства постоянно модернизируются и будут дополняться новыми МП контроллерами (например, на малопотребляющей элементной базе) и перспективными дисковыми МП системами.

Первый набор выполнен на базе однокристалльной ЭВМ серии K1816 в конструктиве E1 (100×160 мм). Он включает в себя два однокристалльных микроконтро-

лера ОКМК-1—02 и ОКМК-1—04. Модули ОКМК предназначены для встраивания в приборы, не рассчитаны на расширение по памяти. Они не имеют системного интерфейса, а для связи с центральной МП системой или другим МП блоком снабжены программным последовательным интерфейсом (для чего используются биты порта и вход прерывания). Модули обслуживают клавиатуру, символьные индикаторы и обрабатывают аналоговую (ОКМК-1—02) или время-импульсную (ОКМК-1—04) информацию. Модули ОКМК используются более чем в 10 приборах и аппаратах, находящихся в серийном производстве. Для обеспечения работы с ЖК-индикатором «бегущая строка» в одноканальном портативном электрокардиографе разработан специализированный микроконтроллер ОКМК-1—06.

Второй набор выполнен в конструктиве E1.2 (100×220 мм) по магистрально-модульному принципу на базе серии K580 с системным интерфейсом И-41М и включает в себя двенадцать основных унифицированных модулей. Набор характеризуется гибкостью, возможностью наращивания, страничной адресацией памяти (до 1 Мбайта), резидентной ОС СР/М с возможностью организации квазидиска, наличием интерфейсов к ЭЛТ (символьного и графического), к бытовому магнитофону, к дисководу и стандартных последовательных и параллельных интерфейсов. Одноплатный микроконтроллер ОКМК-2—02 может также применяться как самостоятельное изделие. Модули второго набора в основном применяются в комплексах доврачебного и диспансерного обследования, многотерминальных системах сбора данных анамнеза, в электрокардиографах с МП управлением и обработкой данных, а также в отладочных прототипных системах ПС-2—80М.

Базовая конфигурация прототипной МП системы ПС-2—80М: ОЗУ (128 Кбайт), ППЗУ (72 Кбайт), три таймера, восемь уровней прерывания, 96 параллельных линий ввода-вывода, два последовательных канала ввода-вывода, интерфейс к ЭЛТ, внешние устройства (монохромный монитор или черно-белый телевизор, полная алфавитно-цифровая клавиатура, бытового магнитофона). Дополнительные внешние устройства — НГМД, принтер, АЦП.

Программное обеспечение — резидентная квазидисковая ОС СР/М, монитор, программатор ППЗУ, тесты.

Размеры ряда моноблоков с блоком питания и вентилятором на 5, 8 и 13 модулей — 280×300×190 мм, 340×300×190 мм, 440×300×180 мм.

Третий набор на базе серии K580 представлен двумя высокоинтегрированными одноплатными контроллерами в конструктиве двойная Европлата: ОКМК-3—03 (233×160 мм), ОКМК-3—04 (233×220 мм).

Первый ориентирован на индикаторы типа «бегущая строка», второй — на работу с экраном на ЭЛТ.

Модуль ОКМК-3—03 представляет собой двухпроцессорную микросистему на базе двух параллельно работающих процессоров K580BM80A и K1816BE35, это позволяет свести к минимуму проблему быстродействия и облегчает процессы программирования и отладки.

Модуль ОКМК-3—04 применяется в системах обработки данных (СОД), представляющих собой персональную микро-систему, оснащенную ОЗУ/ППЗУ квазидисками с резидентной ОС типа СОМ-80.

Конфигурация СОД следующая: блок электроники с блоком питания, черно-белый монитор, полная алфавитно-цифровая клавиатура, матричный принтер.

Четвертый набор (на базе 16-битной серии K1810) предполагается к производству с системным интерфейсом И-41 в конструктиве двойная Европлата. Четвертый набор совместим по ПО с персональным профессиональным компьютером IBM PC/XT, включает в себя два основных модуля: одноплатный микроконтро-

лер ОМК-4-88 с контроллером мини-НГМД и контроллером графического дисплея. Четвертый набор может наращиваться модулями памяти и интерфейсов из второго набора.

ВНИИМПом были разработаны следующие достаточно простые и эффективные средства отладки и тестирования указанных модулей: диагностический тестер для наладки и контроля однокристалльных микроконтроллеров ТОМ-1816 с набором адаптеров; диагностический тестер ДТ-80 для модулей второго и третьего наборов на базе прототипной системы ПС-2—80М и универсального отладочного модуля; программатор-копировщик ППЗУ серий К573.

В качестве программатора биполярных ППЗУ используется серийный программатор 815.

Для организации входного контроля модулей ОКМК и ОКМК-3—03 разработаны простые тестеры входного контроля.

В качестве средств автоматизации разработки и отладки прикладного ПО созданы внутрисхемные эмуляторы и инженерные пульта (консоли) для указанных модульных наборов.

Программное обеспечение, поддерживающее указанные наборы, состоит из: системного ПО для разработки и отладки программ, связи с внешними устройствами и программирование ППЗУ; библиотеки стандартных и прикладных программ; тестового ПО.

В рамках комплекса также разработаны унифицированные блоки питания и конструкции модулей, блоков и корпусов на базе стандарта СЭВ по Евроконструктивам.

Подробное описание модулей, прототипных систем конструкций, программного обеспечения и примеров реализации приборов приведено в работе «Унифицированные микропроцессорные средства медицинской техники» (Труды ВНИИМП.— М., 1987).

Телефон 211-44-83, Москва

Статья поступила 10.9.88

УДК 681.3.06.068

Г. В. Белоусова, И. В. Болмазов, В. Б. Кулаков,
В. В. Несветайлов, Н. А. Федорова

ИНСТРУМЕНТАЛЬНАЯ СИСТЕМА ОТЛАДКИ ПРОГРАММ

Опыт разработки управляющих систем и средств автоматизации научных исследований на ЭВМ, базирующихся на системе команд СМ4 (это мини-ЭВМ СМ4, СМ1420, микроЭВМ «Электроника 60», «Электроника МС 1201», «Электроника 85», а также персональные компьютеры ДВК2М, БК0010), показал, что использование стандартного программного обеспечения для отладки ассемблерных программ не удовлетворяет современному уровню. Особенно устарели отладчики ODT и TODT [1], поставляемые в составе стандартного программного обеспечения, которые не обеспечивают оперативное и удобное внесение изменений в программу в процессе отладки, а информация (команды и данные) отображается в не наглядной форме. Процесс отладки требует высокой квалификации и больших затрат времени программиста.

Исходя из этого был разработан отладчик ODT/X с расширенным набором функций и более «дружественным» интерфейсом. Разработанную систему можно использовать в качестве монитора для специализированных микроЭВМ. В этом случае она обеспечивает более широкий сервис, чем стандартная программа «Пульт» ЭВМ СМ4, «Электроника 60», ДВК2М. Отладчик ODT/X легко перенастраивается под конкретную ОС (ОС РВ, РАФОС [2] и т. д.), а также на работу без ОС и с любой конфигурацией аппаратных средств, имеющей дисплей, клавиатуру и не менее 8 Кбайт ОЗУ или ПЗУ для размещения собственно программ отладчика. Перечень команд системы приведен в таблице. Рассмотрим

Команды отладчика ODT/X

Команда	Синтаксис	Выполняемые функции
D	D[<адр1>] [—<адр2>]	Отображение содержимого памяти с <адр1> по <адр2> в восьмеричном и символьном виде. Если <адр2> не задан, то отображаются 12 строк
L	L[<адр1>] [—<адр2>]	Дизассемблирование программ, начиная с <адр1> по <адр2>, с распечаткой кодов команд. Если <адр2> не задан, то на экран выводится 10 команд
M	M<адр1>, <адр2>, <адр3>	Перемещение содержимого памяти из указанной области [<адр1>, <адр2>] в область, начиная с <адр3>
R	R<n>	Чтение, запись в регистр <n>
G	G[<адр1>] [,<адр2>]	Запуск программы, начиная с <адр1>; <адр2>—точка останова
S	S<адрес>	Запись, чтение содержимого указанной ячейки памяти
A	A<адрес>	Запись в память, начиная с <адреса> команд, вводимых в мнемоническом виде
W	W<адрес>	Запись в память по <адресу> последовательности символов в коде ASCII
V	V<адрес>	Запись в память по <адресу> последовательности символов в коде RADIX50
X	X	Вывод слова состояния с указанием всех признаков, вывод регистров, текущего значения счетчика команд и соответствующей команды
T	T[<n>]	Исполнение <n> команд, адрес первой из которых находится в программном счетчике, и выполнение функции X при исполнении каждой команды
C	C<адрес>	Установить к. т. по <адресу>
E	E[<адрес>]	Удалить к. т. по <адресу>. Если <адрес> не указан, то удалить все к. т.
Q	Q	Выход из отладчика ODT/X
I	I[<устройство>] <файл> [,<тип>] [,<адр zapr>]	Загрузка файла с диска в ОЗУ

Команда	Синтаксис	Выполняемые функции
O	O [〈устройство〉] <файл〉 [〈тип〉], <адр нач〉 , <адр кон〉	Запись программы на диск
B	B	Печать контрольных точек
^P	^P	Параллельный вывод информации на принтер (Запретить/разрешить)

возможности системы ODT/X на примере сеанса отладки.

Пользователь, вызвав отладчик, загружает с его помощью отлаживаемую программу в память ЭВМ. Он может просмотреть программу в кодовом и дизассемблированном виде, где коды операций представлены мнемониками, а адреса операндов вычислены и выводятся их абсолютные значения. Перед запуском программы или части программы на исполнение возможно внесение изменений в программу или данные. После этого следует собственно процесс отладки: задаются контрольные точки останова и программа запускается на исполнение. После ее выхода на останов пользователь просматривает программу и данные, вносит, если необходимо, изменения (в мнемоническом виде, максимально приближенном к мнемонике ассемблерных команд). При необходимости часть команд программы исполняется в пошаговом режиме, отладчик на каждом шаге обеспечивает вывод подробной и наглядной информации о состоянии программы. После отладки части программы ненужные контрольные точки удаляются, а новые назначаются для следующего отлаживаемого модуля.

Эти циклы отладки могут повторяться несколько раз и таким образом за один сеанс работы в отладчике исправляется несколько ошибок в отлаживаемой программе без повторения редактирования, ассемблирования и компоновки исходной программы. Кроме того, возможна разработка и отладка небольших программных модулей непосредственно в отладчике. При необходимости отлаживаемая программа и данные записываются на диск.

В отладчике реализован механизм, который перехватывает прерывание, возникшее из-за ошибки в отлаживаемой пользовательской программе и возвращает управление отладчику. При этом на экран дисплея выводится информация об ошибке. Например, при попытке исполнить несуществующую команду, происходит внутреннее прерывание и обрабатывается подпрограмма отладчика, которая индицирует ошибку и «выходит» в отладчик. В случае заикливания программы пользователя предусмотрен выход из этого состояния по ^C, после чего выводится адрес останова программы пользователя.

В программе ODT/X предусмотрено редактирование входной строки, т. е. с помощью клавиши ЗБ (забой) можно корректировать директиву пользователя. В случае неверной директивы выводится сообщение об ошибке. Можно установить параллельный вывод информации на принтер.

Отладчик состоит из следующих модулей: связь с оператором; прием, обработка директив; служебные подпрограммы (драйверы устройств, выбор параметров директив и т. п.); программы, реализующие директивы.

Структура данных: буфер приема строки (20 байт); область сохранения данных пользователя (26 байт); область рабочих таблиц (1000 байт).

Общий объем программы ODT/X — 7 Кбайт.

При работе с отладчиком можно использовать два варианта компоновки пользовательских программ: отдельным загрузочным модулем (загружаются в оперативную память непосредственно из отладчика); совместно с объектным модулем отладчика (адреса программ и меток можно получить из протокола компоновки).

Пример использования отладчика ODT/X, в котором отлаживается программа вывода текста на экран. Для удобства пример снабжен комментариями, приведенными после точки с запятой:

```
*A1000 ; ввод программы с адреса 1000
1000 TSTB 1777564 ; опрос готовности
1004 BPL 1000 ; дисплея
1006 MOVB (RO)+, 177566 ; вывод символа на дисплей
1012 TSTB (RO) ; проверка на конец
1014 BNE 1000 ; текста
*L1000, 1014 ; дизассемблирование программ
1000 105767 176560 TSTB E, 177564
1004 100375 BPL 1000
1006 112067 176554 MOVB (RO)+, E, 177566
1012 105710 TSTB (RO)
1014 001371 BNE 1000
*W1030 ; ввод текста
1030 * ПРИМЕР РАБОТЫ ODT/X
*S1056 ; запись признака конца текста
1056/177777 0
*D1000, 1060 ; дампы памяти
1000 105767 176560 100375—112067 176554 105710 .Г...7.Л...
1014 001371 000240 000240—000240 000240 000240.....
1030 071160 066551 071145—071040 061141 072157 ПРИМЕР РАБОТ
1044 020171 042117 027524—020130 005015 000000 Ы. ODT/X...
1060 000000
*RO ; занесение в регистр RO адреса
RO/O 1030 ; начала текста
*G1000, 1000 ; запуск программы с адреса
1000, ; задание точки останова по
; адресу 1000
STOP ADDRESS=1000 ; сообщение отладчика о вы-
; ходе на останов
; трассировка одной команды
*T
PSW=4 PRIOR=0 T=0 N=0 Z=1 V=0 C=0 TSTB E, 177564
RO=1030 R1=0 R2=0 R3=0 R4=0 R5=0 SP=1000 PC=1000+1004
*K1016 ; задание контрольной точки
; по адресу 1016
*G ; продолжение исполнения про-
; граммы
ПРИМЕР РАБОТЫ ODT/X ; текст, выведенный програм-
; мой
STOP ADDRESS=1016 ; сообщение отладчика о вы-
; ходе на останов
```

В дальнейшем разработчики системы предусматривают возможность ее усовершенствования: перемещаемость ODT/X; использование символьных меток при работе отладчика, в том числе использование таблиц меток, созданных программами MACRO, LINK [3]; восстановление исходного текста кодовых программ в формате, обеспечивающем дальнейшее редактирование и ассемблирование; версия с улучшенной интерактивностью, использующая многооконный режим работы.

140700, Московская область, г. Шатура,
пр. Ильича, д. 53, кв. 45, Болмазов И. В.

ЛИТЕРАТУРА

1. Синглер М. МиниЭВМ PDP-11: Программирование на языке ассемблера и организация машины.— М.: Мир, 1984.
2. Операционная система СМ ЭВМ РАФОС/Л. И. Валикова, Г. В. Вигдорчик, А. Ю. Воробьев и др.— М.: Финансы и статистика, 1984.
3. Гилл А. Программирование на языке ассемблера для PDP-11.— М.: Радио и связь, 1983.

Статья поступила 26.01.1988

ПРОСТОЙ ПРОГРАММАТОР ДЛЯ ПЗУ ТИПА К573РФ3

В большинстве серийно выпускаемых программаторов для ПЗУ и в оригинальных опубликованных схемах [1—5] отсутствует возможность программирования ПЗУ К573РФ3 емкостью 4К×16 [6].

Предлагаемый простой (100×85×15 мм) программатор (см. рисунок) для этих БИС содержит девять микросхем низкой степени интеграции. Управление программатором осуществляется магистралью МПИ, вследствие чего он может быть подключен, например, непосредственно к колодке пользователя (КУ) на плате НМС 11100 в микроЭВМ ДВК 1... ДВК 4.

Программатор можно обслуживать операционной системой микроЭВМ с помощью простейшего драйвера, обеспечивающего интервал в 50 мс между посылками слов данных для записи системными средствами таймаута или путем транзитной обработки прерываний от таймера.

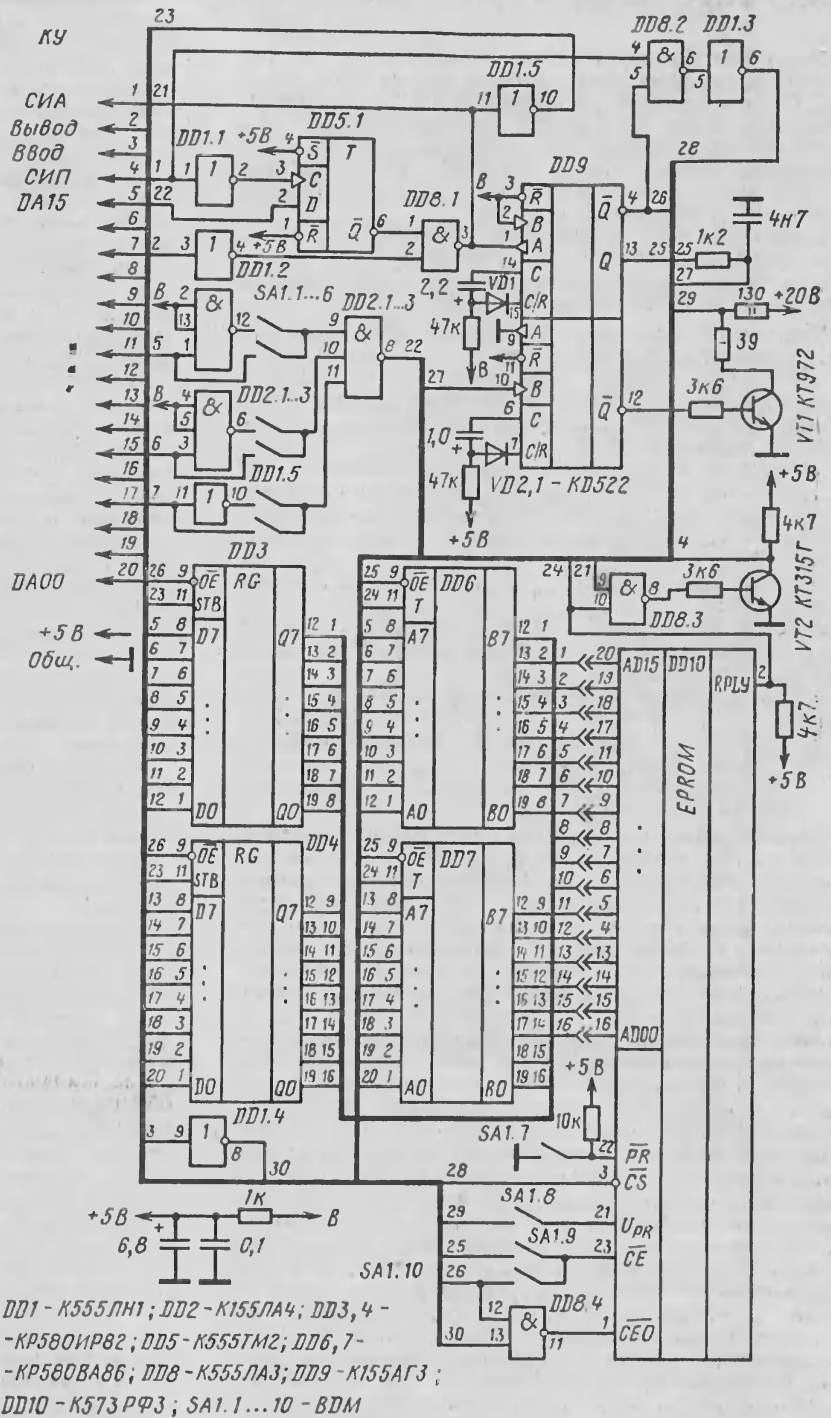
Режимы программирования БИС (чтение и запись данных, запись кода банка) и номер банка, в котором производится программирование, устанавливаются переключателями А1.1...А1.9. Транзистор VT2 необходим для инвертирования и согласования с сигналом СИП на плате НМС 11100. Диоды VD1,2 увеличивают надежность работы одновибратора DD9.

Телефон 246-05-80, Москва, с 10 до 17 ч.

ЛИТЕРАТУРА

1. Дианов А. П., Щелкунов Н. Н. Модуль программирования микросхем ПЗУ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 80—83.
2. Лукьянов Д. А. Схемотехника универсальных программаторов ПЗУ // Микропроцессорные средства и системы. — 1985. — № 3. — С. 84—88.
3. Барabanов А. Б., Турунов Н. Г. Простой программатор // Микропроцессорные средства и системы. — 1985. — № 2. — С. 84—85.
4. Найденов А. В., Романенков В. А. Программатор ПЗУ на базе микроЭВМ «Электроника К1-20» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 34—39.
5. Жихарев В. И. Программатор на базе микроЭВМ «Электроника ДЗ-28» // Микропроцессорные средства и системы. — 1986. — № 5. — С. 40—43.
6. БК0.348.648 ТУ. Микросхемы К573РФ3. Технические условия.

Сообщение поступило 28.10.87



Принципиальная схема программатора для ПЗУ типа К573РФ3

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММ ДЛЯ МИКРОПРОЦЕССОРА KM1810VM86

Инструментальный комплекс базируется на локальной вычислительной сети, работающей на ЭВМ серий ЕС, СМ и микроЭВМ «Электроника ДЗ-29». Программная часть комплекса включает кроссассемблер для МП KM1810VM86, компоновщик объектных модулей (оба компонента реализованы на СМ ЭВМ), программу поддержки МП, встроенную в монитор интеллектуального терминала, и диалоговый отладчик.

Аппаратная часть отрабатывает интерфейс между шиной ДЗ-28 и шиной МП, обеспечивая доступ к памяти МП, а также обмен с ним через порты ввода-вывода с использованием прерывания по вектору.

Технология разработки прикладной программы для МП включает следующие шаги:

написание и отладку программы либо на ДЗ-28 с последующей передачей ее на СМ ЭВМ, либо непосредственно на СМ ЭВМ с ее средствами ввода и редактирования программ;

трансляцию программы, исправление ошибок; компоновку оттранслированной программы с созданными ранее программами для получения абсолютного модуля, настроенного на конкретный указанный адрес памяти МП;

передачу абсолютного модуля отладчика из ДЗ-28 в память МП;

передачу абсолютного модуля созданной программы из СМ ЭВМ в ДЗ-28 средствами монитора интеллектуального терминала;

пересылку абсолютного модуля из памяти ДЗ-28 в память МП с использованием программы поддержки МП в ДЗ-28 и аннотированной части обеспечения интерфейса ДЗ-28 с KM1810VM86;

запуск программы в МП и отладку ее с помощью диалогового отладчика.

Кроссассемблер реализован в среде ОС РВ на языке Паскаль. При его запуске определяется необходимость создания объектного кода ассемблируемой программы и листинга. Программирование ведется в мнемонике команд МП с использованием символических имен переменных; констант в различных системах счисления; команд резервирования памяти, управления счетчиком адреса, определения строковых данных; команд условной трансляции.

Листинг содержит значения счетчика адреса, сгенерированные коды команд МП, исходный текст программы. Диагностические сообщения об ошибках вставляются в некорректные строки программы непосредственно в листингах трансляции.

Объектный модуль содержит информацию об областях программы, глобальных и локальных метках, о переместимых адресах и сгенерированные коды. Вся эта информация необходима для компоновки отдельно ассемблированных модулей в абсолютный модуль, настроенный на конкретный адрес памяти МП. Эту функцию выполняет компоновщик, также реализованный в среде ОС РВ на языке Паскаль, результатом работы которого, кроме абсолютного модуля, является карта распределения памяти с указанием адресов переменных и таблица перекрестных ссылок. Полученный абсолютный модуль сохраняется в файле на магнитном диске.

Средства поддержки МП, встроенные в монитор последовательного терминала на ДЗ-28, позволяют выполнять в диалоговом режиме следующие функции:

передачу абсолютного модуля или части его из ДЗ-28 в память МП по заданному адресу;

получение в ДЗ-28 содержимого памяти МП с заданного адреса;

выдачу шестнадцатеричного дампа памяти МП на дисплей;

запуск программы МП с указанного адреса;

сброс МП, захват и освобождение его шин.

Функции обмена с памятью могут осуществляться без останова и сброса МП, что позволяет динамически отслеживать работу программы, считывая в ДЗ-28 область данных и область трассировки, анализировать их и передавать в память МП новые значения данных. Этот режим, в первую очередь, необходим для отладки программ, не ориентированных на диалоговый режим работы.

Для разработки и отладки программ, ориентированных на диалог, предусмотрена возможность побайтного обмена через порты ввода-вывода и соответствующий вектор прерывания между МП и ДЗ-28. Этот режим поддерживается в мониторе интеллектуального терминала средствами обмена строками данных между дисплеем ДЗ-28 и МП. Помимо этого есть возможность сохранять в памяти ДЗ-28 все выведенные из МП данные для дальнейшего анализа при отладке программы.

Диалоговый отладчик предоставляет возможность пошаговой отладки прикладных программ, загруженных в память МП. Диалог ведется с терминала ДЗ-28 и позволяет задавать точки останова в пошаговом режиме, выводить содержимое всех или только указанных регистров МП, просматривать и изменять содержимое регистров и памяти, переносить управление на заданную точку программы. Адреса могут передаваться в абсолютном формате или с помощью базового адреса и смещения, что удобно при отладке по листингам отдельно ассемблированных модулей. При передаче строки с дисплея ДЗ-28 в МП драйвер соответствующего порта анализирует и определяет ее назначение. Если она предназначена прикладной программе, то отладчик не активизируется. В противном случае запускается отладчик и весь диалог с этого момента и до выхода идет с ним.

Незначительные доработки позволили выполнить прямое согласование интерфейсов «Печать» ДЗ-28 и устройства матричной печати типа DZM-180. Объем оперативной памяти ДЗ-28 128 Кбайт.

Интерфейсный модуль (ИМ) состоит из преобразователя параллельного кода интерфейса ДЗ-28 в ИРПС для связи с дисплеем 15ИЭ-00-013 и СМ ЭВМ и согласователя интерфейсов ДЗ-28 с внутренней шиной МП. Скорость обмена между ОЗУ ДЗ-28 и МП должна быть максимальной, поэтому интерфейсный модуль отрабатывает команды быстрой передачи данных, обеспечивая скорость передачи до 200 Кбайт/с. Со стороны ДЗ-28 требуется программная установка начального адреса ОЗУ и захват шин МП. Отработка передаваемых или принимаемых данных выполняется автоматически.

Модуль имеет также канал обмена управляющей информацией между ДЗ-28 и МП со своим вектором прерывания. Канал может быть задействован в динамике по ходу выполнения программ в ДЗ-28 с прерыванием от МП и наоборот. Средства установки приоритетов, их анализ и организация цикла прерывания — стандартные для МП, работающего совместно с K589ИК14. Кроме того, ИМ отрабатывает программный перезапуск программ МП, захват и снятие захвата шины МП по соответствующим стандартным диаграммам.

Пульс индикации состояния и управления МП включен в себя индикаторы 32 портов вывода и один порт ввода информации. Введены также индикаторы шины адреса и данных, адреса останова по содержимому адресной шины команд или данных и вектора прерывания, задаваемого с пульта. Функции управления — стандартны для устройств подобного типа: пошаговый и автоматический режимы, останов по неготовности, захват шин, перезапуск и прерывание по заданному вектору.

Вся аппаратная часть выполнена в виде настольного комплекса. Допустимое удаление от инструментальной ЭВМ типа СМ ЭВМ определяется возможностями стандартного интерфейса ИРПС. Интерфейсный модуль потребляет 2 А от сети с напряжением 5 В, а пульт управления и индикации — 3 А.

420012, Казань, 12, ул. Комлева, 20, КФ ИПИАН

Статья поступила 08.10.86

УДК 681.3.06

А. М. Белов, О. В. Белов, Л. Л. Муренко,
А. С. Шестиалтынов

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ОТЛАДКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В КОМПЛЕКСЕ КРОССПРОГРАММ «ЭЛЕКТРОНИКА МИКРОС 1810»

В статье обобщается опыт использования программно-логических моделей в комплексах кросспрограмм серии «Электроника МИКРОС».

Основные функции ПЛМ данного кросс-средства: определение конфигурации микропроцессорного устройства (наложение ограничений на доступ к памяти, портов ввода-вывода и т. п.);

загрузка в виртуальную память отлаживаемых модулей;

имитация выполнения команд с указанного адреса в автономном или пошаговом режиме, прерываний МПУ; даммирование памяти и программно-доступных регистров (ПДР);

изменение значений ячеек памяти и ПДР; протоколирование процесса отладки.

Дополнительные функции:

останов имитации выполнения команд в контрольных точках (по обращению к конкретному адресу) и по различным условиям (операции ввода-вывода, значению результата операции, ячейки памяти, конкретного ПДР); возможность получения дампа указанных ячеек памяти и ПДР за несколько шагов до определенного события (прерывания, останова по условию или контрольной точке);

оценка выполнения программных компонентов в реальном масштабе времени;

проведение символической отладки ПО, при которой используются имена и метки, описанные в исходных текстах (например, при даммировании памяти МПУ), и воспроизводятся оригинальные записи выполняемых ПЛМ команд.

На емкость внешних запоминающих устройств накладываются минимальные ограничения, так как в комплексе кросспрограмм «Электроника МИКРОС 1810» в процессе отладки не происходит физического получения загрузочных модулей и выделения специальной области под виртуальную память МПУ. Загрузочный модуль и виртуальная память представляются совокупностью объектных модулей созданных при ассемблировании исходных текстов. Изменяя внутреннее содержание файлов объектных модулей без создания новых, на внешних запоминающих устройствах инструментальной ЭВМ создается загрузочный модуль и осуществляется его загрузка в виртуальную память.

Объем ПО отлаживаемого с помощью ПЛМ, зависит только от емкости внешних запоминающих устройств, хранящих файлы объектных модулей.

Программно-логическая модель данного кросс-средства выполняет все функции редактора связей и загрузчика.

По завершении отладки возможно физическое получение загрузочного модуля в стандартном виде.

Телефон 468-13-70, Москва

Сообщение поступило 19.08.87

УДК 681.325.5 : 681.326

В. В. Кольцов, К. А. Пархоменко, В. А. Прошин,
А. В. Вершинин

СИСТЕМА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ И ОТЛАДКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КТС ЛИУС-2

Для создания и сопровождения микропроцессорных АСУТП на базе КТС ЛИУС-2 разработан программно-аппаратный отладочный комплекс на базе ПЭВМ «Искра 226» (кроссовая ЭВМ). Он предназначен для разработки и отладки ПО входящих в состав АСУТП микропроцессорных блоков (МП-блоков) контроля и диагностики функциональных элементов КТС ЛИУС-2.

Общая характеристика отладочного комплекса:

скорость трансляции до 150 команд/с;
объем отлаживаемых программ в пределах одного МП-блока до 64 Кбайт;
скорость обмена кроссовой ЭВМ с МП-блоком — до 20 Кбайт/с;
максимальное число одновременно включаемых блоков — 7;
отладка ПО в режиме реального времени;
возможность подключения к МП-системе в период промышленной эксплуатации с сохранением ее работоспособности;
одновременная отладка ПС нескольких МП-блоков;
контроль и диагностика функциональных элементов МП-системы в периоды разработки и эксплуатации.

Состав отладочного комплекса:

система подготовки ПО предназначена для ввода и корректировки исходных текстов программ и документации, ведения архивов на магнитных дисках, трансляции и компоновки программных модулей форматирования технической и программной документации. Работа ведется в диалоговом и пакетном режимах. Механизм командных файлов позволяет автоматизировать процесс генерации ПО. Использование макросредств существенно расширяет возможности языка ассемблера МП К580ИК80;

интерактивный кроссовый отладчик (DEBUG) позволяет контролировать состояние подключенных к кроссовой ЭВМ МП-блоков. Каждая адаптерная плата имеет свой номер (от 1 до 7), используемый в качестве идентификатора при обращении. При одновременной работе с несколькими МП-блоками один из них находится в интерактивном режиме (при отладке доступны все директивы отладчика), а остальные — в режиме слежения;

плата преобразователя интерфейса (ПИ);

набор адаптерных плат (АП) с режимом прямого доступа к магистрали МП-блока и безусловным приоритетом по отношению к процессорному модулю при обращении (операциях отладки: останов МП, пошаговое выполнение программы, чтение и модификация регистров ОЗУ и портов ввода-вывода).

Кроссовый отладчик предоставляет пользователю следующие средства отладки: загрузку ПО в ОЗУ МП-блока, сброс МП-блока, пошаговую отладку ПО, назначение адресов останова МП, чтение и модификацию ресурсов МП-блока (регистры, программный счетчик, порты ввода-вывода, шины запроса прерываний), запуск и останов МП, динамическое отображение содержимого ОЗУ любого МП-блока. Для быстрой проверки исправности плат, используемых при отладке, предназначены раздел тестов. Они выполняются процессором МП-блока или кроссовой ЭВМ. По назначению тесты делятся на пять групп: элемента управления, ОЗУ, ПЗУ, элементов ввода-вывода, ПИ и АП. Их выбор и отображение результатов работы производится через кроссовую ЭВМ.

Отладочный комплекс эксплуатируется с 1985 г., закончена разработка на базе ПЭВМ «Роботрон 1715», 344017, Ростов-на-Дону, пл. Народного ополчения, 2, Ростовский институт инженеров железнодорожного транспорта, ОНИЛ «МИУС»; тел. 59-51-53, 31-12-62

Сообщение поступило 06.10.87

ЛОГИЧЕСКИЙ АНАЛИЗАТОР

Широкое распространение микропроцессоров (МП) привело к появлению новых измерительных приборов — логических анализаторов (ЛА), регистрирующих изменения цифровых сигналов во времени и отображающих эти изменения в виде, удобном для анализа работы аппаратных средств или ПО [1—2].

Такие качества ЛА, как многоканальность, наличие быстродействующих блоков памяти для регистрации цифровых данных, развитая система управления запуском процесса регистрации, наличие различных режимов тактирования при записи, выбор наиболее удобной формы визуального представления зарегистрированных данных, отображение на индикаторе прибора вспомогательных указателей и служебной информации, облегчающих анализ отображаемой информации, существенно упростили диагностику и ремонт сложных цифровых устройств и систем.

Основная задача описанного ниже ЛА (класс анализаторов временных диаграмм) — анализ логических состояний во времени. Анализаторы временных диаграмм расширяют возможности обычных осциллографов при наблюдении единичных или очень редко повторяющихся событий, временных зависимостей между многими сигналами и состояний сигналов перед запуском. Они ориентированы только на цифровые сигналы и наиболее эффективны при анализе работы аппаратных средств цифровых устройств и систем.

Логический анализатор [3] обеспечивает следующие возможности по регистрации отображению и обработке цифровой информации об изменении логических состояний контролируемой схемы (см. 4-ю стр. вкладки):

- запуск процесса регистрации по 1-му или (n+1)-му обнаружению в контролируемом потоке данных кодового слова запуска (n=1...9);
- регистрацию данных, предшествующих кодовому слову запуска на заданное число тактов записи;
- регистрацию данных в режиме сравнения массивов, при этом в одном цикле записи процесс регистрации запускается дважды по двум последовательным или (n+1)-м обнаружениям в контролируемом потоке данных кодового слова запуска;
- регистрацию данных в перечисленных режимах по фиксированным значениям тактов внутренней или внешней синхронизации;
- отображение зарегистрированной информации в виде временных диаграмм;

- возможность растяжки и сжатия отображаемой информации;
- возможность попеременного отображения соответствующих участков двух массивов, зарегистрированных в режиме сравнения;
- отображение двух курсоров;
- отображение служебной информации о параметрах выбранных режимов (записи и отображения);
- вычисленные расстояния между курсорами в тактах записи и отображение вычисленной величины на экране;
- задание и поиск в массиве зарегистрированной информации новых квалифицирующих слов.

Алгоритм работы (рис. 2) МП блока управления содержит три модуля: отображения, обработки прерываний и стартовый.

Стартовый модуль иницируется при включении ЛА, запрещает прерывания программы, программирует порты ввода-вывода, тестирует внутреннее ОЗУ МП блока управления, задает типовой режим записи и отображения (по практическим данным о наиболее частой встречаемости), формирует в участках хранения отображаемого текущего кадра, маски отображения начальной «картинки».

Основные технические характеристики логического анализатора

Число каналов	8
Объем, Кбайт	3
ПЗУ	2
ОЗУ	1
быстродействующего ОЗУ	
Частота тактов, МГц	20; 0.2; 0.02
внутренней синхронизации	20
внешней синхронизации	20
Цифровая задержка запуска процесса регистрации на число появлений слова запуска	1...9
Разрядность кодового слова запуска, бит	0...8
Число тактов регистрации, предшествующих такту обнаружения слова запуска	0...1023
Масштабы при отображении	1:2, 1:5, 1:10
сжатия	4:1
растяжки	65
Потребляемая мощность, Вт	400×325×190
Габаритные размеры, мм	11
Масса, кг	

Структурная схема ЛА (рис. 1) включает в себя МП блок управления на базе КР580ИК80А [4], содержащий модуль ПЗУ—ОЗУ; блок отображения на основе матричной индикаторной панели ИМГ1-01 [5]; блок ввода параметров, обнаружения кода запуска и управления записью; быстродействующее ОЗУ; 1-й и 2-й буферные регистры.

Клавиатура для управления ЛА находится в блоке ввода параметров.

Затем модуль отображения поэлементно развертывает на матричной индикаторной панели маску отображаемой «картинки». После разворачивания одного кадра программно разрешаются прерывания. При наличии запроса на прерывание (формируется в блоке ввода параметров при нажатии клавиши клавиатуры) происходит переход программы на выполнение модуля обработки прерываний, иначе программа

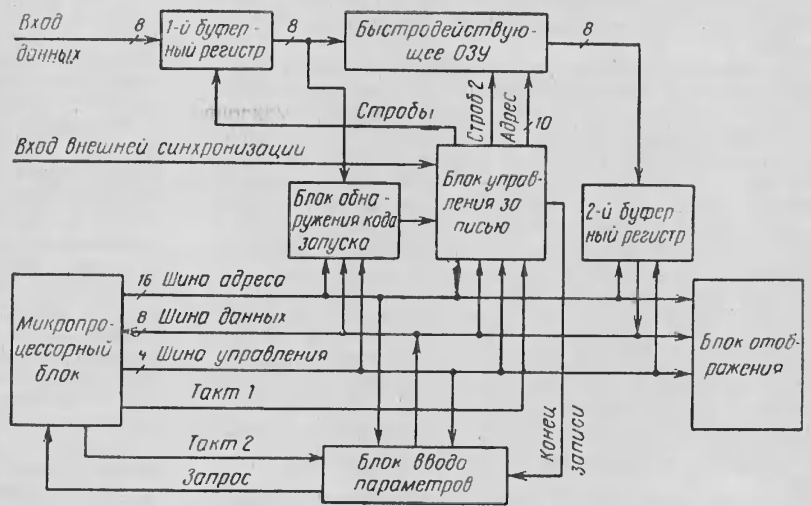


Рис. 1. Структурная схема логического анализатора

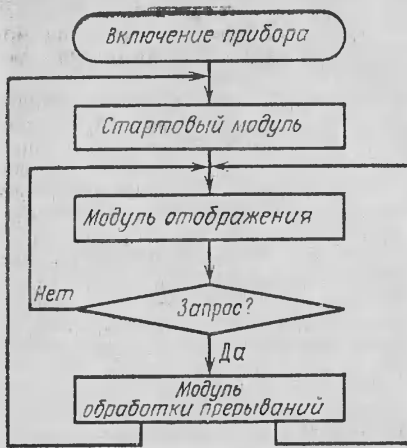


Рис. 2. Алгоритм работы логического анализатора

вновь, запретив прерывания, возвращается на начало модуля отображения.

Модуль обработки прерываний, программно запретив новые прерывания, осуществляет предписываемые нажатой клавишей операции либо по изменению режимов записи или отображения, либо по перезаписи данных из быстродействующего ОЗУ через 2-й буферный регистр во внутреннее ОЗУ МП блока, либо по обработке зарегистрированных данных и затем возвращает выполнение на начало модуля отображения, а при нажатии клавиши «Сброс» — на начало стартового модуля.

При таком построении алгоритма маска отображаемой «картинки» на экране матричной индикаторной панели разворачивается непрерывно с кадровой частотой, большей минимально допустимого значения (25 Гц), и исключает прерывания во время разворачивания очередного кадра. Это улучшает качество изображения.

Заданный режим регистрации данных реализуют блоки обнаружения кода запуска и управления записью. Это обеспечивает работу ЛА одновременно в режимах регистрации данных и отображения предыдущей записи.

Быстродействующее ОЗУ и сверхбыстродействующий 1-й буферный регистр, работающие под управлением специализированного блока управления записью, позволяют регистрировать данные на частотах в десятки мегагерц. Это крайне важно для анализа работы аппаратных средств. Специализированный блок обнаружения кода запуска дает возможность на частоте регистрации контролировать появление кодового слова запуска во входном потоке данных.

Выбор индикаторной матричной панели ИМГГ-01 в качестве ус-

ройства отображения ЛА, ввиду малых габаритов и простоты управления цифровыми методами, обеспечил малые габариты, массу и простоту прибора в целом. Характер отображаемой информации (временные диаграммы в виде двух логических уровней и буквенно-цифровая информация) хорошо согласуется с дискретной структурой панели. Это положительно сказывается на восприятии отображаемой «картинки». Небольшое количество точек отображения по строке (100) компенсируется в описываемом ЛА возможностью изменения масштаба отображения и сдвига отображаемого «окна» по массиву зарегистрированной информации. Все манипуляции пользователя с органами управления прибора вызывают мгновенный отклик на индикаторе.

При реализации ЛА использованы микросхемы серий К132, К155, К531, К565, К573, К580. Время цикла записи-считывания микросхем памяти КР132РУ4А и КР133РУ7 — 47 и 30 нс. Это обеспечивает частоту регистрации данных в 20 МГц. Модульный принцип построения структурной схемы определяет свободу конструкторского исполнения прибора, что дает возможность изготовить его в условиях лаборатории.

Опытный образец ЛА широко использовался для ремонта устройств

ввода-вывода и при отладке вновь разрабатываемой аппаратуры цифровой обработки сигналов. При этом сроки проведения ремонта и отладки значительно сократились.

Телефон 546-95-01, доб. 38-64, Москва

ЛИТЕРАТУРА

1. Богородицкий А. А. Комплекс аппаратуры для типового технологического процесса контроля при производстве РЭА // ПСУ.— 1985.— № 6.— С. 23—25.
2. Бабкин П. А. и др. Программируемый активный анализатор логических состояний для эксплуатационного обслуживания цифровых систем // УСИМ.— 1982.— № 3.— С. 15—19.
3. А. с. № 1259267 СССР. Логический анализатор / В. М. Андреев, С. П. Леухин.— Оpubл. 1986, Бюл. № 35.
4. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.
5. А. с. 1336097 СССР. Устройство для отображения информации / В. М. Андреев, С. П. Леухин, В. И. Писларь.— Оpubл. 1987, Бюл. № 33.

Статья поступила 04.05.87

УДК 681.322

В. А. Дмитриев, А. В. Бедарев, В. С. Гравов, А. Д. Бурнашев, С. Н. Сапо

СТЕНД ДЛЯ ИЗУЧЕНИЯ ВРЕМЕННЫХ СООТНОШЕНИЙ СИГНАЛОВ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Разработано устройство для изучения сигналов микросхем МП КР580ВМ80, тактового генератора КР5801Ф24, системного контроллера КР580ВК28. Восемь сигналов постоянно отображаются на экране однолучевого осциллографа при непрерывном выполнении одной команды, произвольно выбранной из системы команд микропроцессора (рис. 1).

Стенд позволяет изучить процесс выполнения выбранной команды в поцикловом режиме с индикацией типа (кода) машинного цикла, состояния шины данных, номера цикла, управляющих сигналов на светодиодных линейках стенда.

Устройство имеет режим загрузки, в котором во внутреннее ОЗУ стенда побайтно записывается машинный код выбранной команды. Для набора байта используется тумблерный регистр.

Стенд представляет собой автономный прибор с габаритными размерами 330×280×50 мм и внешним источником питания. На передней панели расположены органы управления и индикации, гнезда для подключения осциллографа. Электриче-

ская схема стенда реализована 27 микросхемами серий КР580, К589, К155, К572 на плате размерами 165×140 мм. В течение двух лет на стенде проводятся лабораторные работы по кафедре вычислительной и измерительной техники Московского института нефти и газа имени И. М. Губкина студентами факультета Автоматики и вычислительной техники.

Функциональная схема стенда состоит из трех узлов (рис. 2): исполнения команды (2), управления и индикации (1), вывода отображаемых сигналов на осциллограф (3).

Узел исполнения команды обеспечивает последовательное многократное считывание всех байтов команды из ОЗУ стенда и ее выполнение микропроцессором в непрерывном режиме или по машинным циклам с запуском от кнопки. Цикличность выполнения команды реализуется с помощью внешнего счетчика адреса ОЗУ. При выполнении каждого машинного цикла команды содержимое счетчика инкрементируется сигналом STSTB, для считывания следующего байта команды или обмена данными с ОЗУ. После окончания выполнения

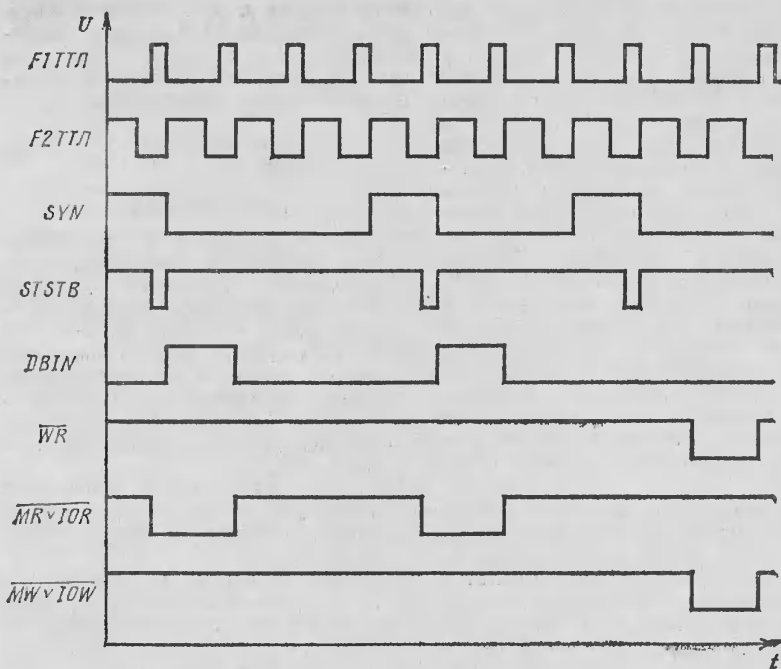


Рис. 1. Временная диаграмма сигналов МП системы при выполнении команды OUT N

команды счетчик сбрасывается, вызывая повторное считывание той же команды. Объем ОЗУ (8×8 бит) позволяет как записать код любой команды, так и зафиксировать результаты обмена данными с микропроцессором при ее выполнении. Счетчик снабжен трехстабильным буфером для отключения в режиме загрузки ОЗУ.

Узел управления и индикации используется для предварительной ручной записи в ОЗУ стэнда кода команды и необходимых данных; управления переключением режимов работы; вывода на индикацию при загрузке или поцикловом исполнении команды адреса ОЗУ или номера машинного цикла, содержимого ячейки ОЗУ или состояния шин данных микропроцессора, кода типа машинного цикла (слова PSW процессора).

Для переключения шин микропроцессора в пассивное состояние при загрузке ОЗУ используется входной сигнал микропроцессора HOLD и выходной HLDA, активирующий тристабильные буферы тумблерных регистров.

Узел вывода отображаемых сигналов на осциллограф синхронизирует запуск развертки осциллографа с началом исполнения команды, последовательно коммутирует восемь отображаемых сигналов с периодом, равным времени выполнения команды, смещая их по вертикали. Управление работой элементов этого узла осуществляет трехразрядный двоичный счетчик строк развертки, использующий сигнал SYNC&D5 в качестве счетного. Коды с выхода счетчика, циклически повторяющиеся через восемь счетных импульсов, поступают на адресные входы мультиметра, управляя последовательным прохождением на младший вход цифро-аналогового преобразователя каждого из исследуемых сигналов на время выполнения одной команды. Одновременно код счетчика строк поступает на входы трех старших разрядов ЦАП, формируя на его выходе восемь различных постоянных уровней (строк развертки на осциллографе для каждого исследуемого сигнала). Выходной сигнал ЦАП подается на вход вертикального отклонения луча осциллографа. Для его синхронизации используется сигнал SYNC&D5.

Принцип устройства стэнда может быть применен для создания устройств изучения временных диаграмм микропроцессоров, имеющих внешние сигналы начала исполнения каждой команды. Можно удвоить число сигналов, наблюдаемых на экране, увеличив разрядность счетчика строк и мультиметра до 4 бит.

117917, Москва, Ленинский просп., 65, МИНГ им. Губкина И. М., кафедра вычислительной и измерительной техники; тел. 930-90-09

Статья поступила 20.11.87

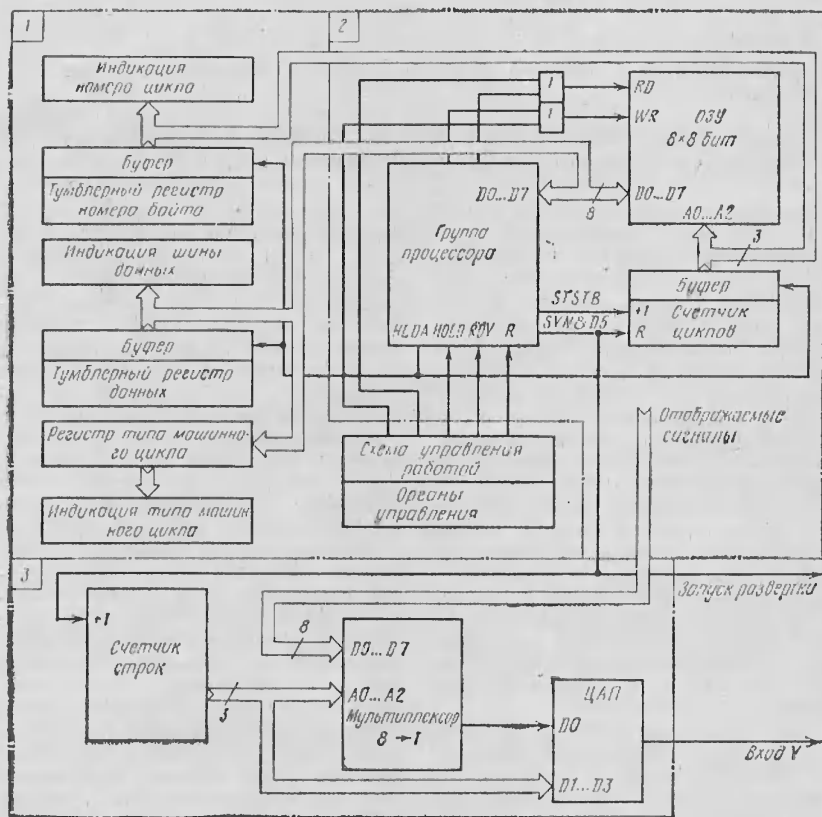


Рис. 2. Функциональная схема устройства

ТЕКСТОВЫЙ ЭКРАННЫЙ РЕДАКТОР SED ДЛЯ ДИСПЛЕЯ «ЭЛЕКТРОНИКА МС 7401»

Использование дисплея МС 7401 с мини-ЭВМ типа СМ4 и микроЭВМ типа «Электроника 60» дает возможность дополнить возможности по отображению графической информации. Однако в операционных системах (ОС) ДВК и РАФОС пока нет соответствующего экранного редактора. Известный экранный редактор SCREEN не подходит для дисплея МС 7401. В составе ОС РАФОС поставляются редакторы К2000, К15 и К521, являющиеся версиями экранного редактора SED* для дисплеев типа ВТА-2000, 15ИЭ-013 и VDT-52129.

Разработанный редактор, названный К7401, выполнен на основе редактора К521, наиболее близкого по системе команд к дисплею МС 7401. В редакторе К7401 табуляция при выводе на экран заменяется пробелами, что снимает несогласованность исходной установки табуляторов дисплея МС 7401 и редактора К521; с использованием полного набора символов введены команды вставки в редактируемый текст восьмичисловых кодов 16 и 17, соответствующих переводу дисплея в русский и латинский регистры; для более удобной работы с протоколом на гибких магнитных дисках (ГМД) уменьшен размер файла протокола редактирования с 30 до 10 блоков, а размер буфера протокола увеличен с 16 до 128 байт; для исполнения часто встречающихся команд дополнительно к назначенным последовательностям* назначены свободные функциональные клавиши клавиатуры дисплея МС 7401; изъяты команды медленного просмотра файла вперед и назад; разработан справочный файл К7401.HLP.

С дисплеем 15ИЭ-013 редактор К7401 работает без изменений. При этом некоторые командные ESC-последовательности необходимо набирать раздельно.

Редактор сконструирован как для работы с процессорами, не имеющими расширенного набора инструкций (размер 80 блоков), так и для процессоров, исполняющих EIS, FIS-инструкции (размер 51 блок). Редактор К7401 работает под управлением FB-и TS-мониторов в ОС ДВК и РАФОС. Для получения полного набора символов русского и латинского алфавитов при работе в ОС RT-11 необходима коррекция монитора для распознавания команд переклочки русского и латинского регистров.

Материалы по описанной версии редактора К7401, включающие исход-

КОМАНДЫ СИМВОЛ	ДОПОЛНИТЕЛЬНАЯ ЦИФРОВАЯ КЛАВИАТУРА				КОМАНДЫ СИМВОЛ	ДОП. КЛАВИАТУРА	
	ПФ1..... команды	ПФ2..... макро- команды	ПФ3повто- рить пос- ледн. макр.	ПФ4 =ПФ1 F		~\~ Вниз на 1 экран = ПФ1 курсор вниз	[^] Вверх на 1 экран =ПФ1 курсор вверх
ПФ1 N ПФ1 ? ПФ1 E= IS ПФ1 D ПФ1 (УПР/А) ПФ1 A ПФ1 Z ПФ1 I=<, > ПФ1 O ПФ1 N=ВВОД ПФ1 V ПФ1 R ПФ1 T ПФ1 S ПФ1 U ПФ1 M ПФ1 B ПФ1 P ПФ1 F ПФ2 (ИМЯ) ПФ3	7 удалить символ	8 вставить пробел	9 поиск по файлу		УПР/L ВЗ(...) <1> <2> <3> <4> <5> <6> <7> <8> <9> <0> <.>		\ Обновление экрана=AP2 E Курсор вверх Курсор вниз Курсор влево Курсор вправо
(УПР/С) (А/Б): нажать клавишу (Б) при нажатой клавише (А) (...) набирается на основной клавиатуре, <...> - на дополнительной КОМАНДЫ	4 удалить до начала строки	5 удалить всю строку	6 удалить до конца строки	ПФ1 I =ПФ1 N Ввести буфер и вести следующую часть файла			
ПФ1 N: вывод справочной информации ПФ1 ? : вывод состояния редактора ПФ1 E: выход с созданием файла .ВАК ПФ1 D: выход без создания файла .ВАК ПФ1 A: в начало буфера ПФ1 Z: в конец буфера ПФ1 I: ввести в буфер следующую стр. ПФ1 O: вывести буфер в выходной файл ПФ1 N: вывести буфер и ввести след. стр. ПФ1 V: режим вставки символов ПФ1 R: режим замены символов ПФ1 T: метка первой сохраняемой строки ПФ1 S: вывести строки от метки T до курсора в файл сохранения ПФ1 U: вставить из файла сохранения ПФ1 M: определить макрокоманду ПФ1 B: считать все макро из макрофайла ПФ1 P: записать все макро в макрофайл ВС - в начало строки, ИЗ - в конец строки.	ПФ1 F (ТЕКСТ) (ВК) <.> : поиск (ТЕКСТ) в буфере ПФ1 F (ТЕКСТ) (ВК) <9> : поиск (ТЕКСТ) по файлу (СИМВОЛ) : вставка/замена символа ТАБ: вставка табуляции ЗБ : удаление предыдущего символа ВК : вставка строки ПС : влево на табуляцию ВШ : вправо на табуляцию (УПР/Л) : вставка перевода формата М А К Р О К О М А Н Д Ы (ИМЯ) - один символ Ввод макро ПФ1М(ИМЯ) (ВК) [КОМАНДЫ] (УПР/А) ПФ2(ИМЯ) : выполнить макрокоманду (ИМЯ) ВЗ(ЧИСЛО) ПФ2(ИМЯ) : выполн. (ИМЯ) (ЧИСЛО) раз ПРОТОКОЛ ИРОВАНИЕ Если ввод имени редактируемого файла оканчивается: (ВК) - ведется протокол, (ПС) - протокол не ведется, (ВШ) - восстанавливается протокол.						

(УПР/С) прерывание макрокоманд; (УПР/У) отмена строки при вводе имени файла. (А/Б): нажать клавишу (Б) при нажатой клавише (А)

(...) набирается на основной клавиатуре, <...> - на дополнительной
КОМАНДЫ
ПФ1 F (ТЕКСТ) (ВК) <.> : поиск (ТЕКСТ) в буфере
ПФ1 F (ТЕКСТ) (ВК) <9> : поиск (ТЕКСТ) по файлу (СИМВОЛ) : вставка/замена символа
ТАБ: вставка табуляции
ЗБ : удаление предыдущего символа
ВК : вставка строки
ПС : влево на табуляцию
ВШ : вправо на табуляцию
(УПР/Л) : вставка перевода формата
М А К Р О К О М А Н Д Ы
(ИМЯ) - один символ
Ввод макро ПФ1М(ИМЯ) (ВК) [КОМАНДЫ] (УПР/А)
ПФ2(ИМЯ) : выполнить макрокоманду (ИМЯ)
ВЗ(ЧИСЛО) ПФ2(ИМЯ) : выполн. (ИМЯ) (ЧИСЛО) раз
ПРОТОКОЛ ИРОВАНИЕ
Если ввод имени редактируемого файла оканчивается: (ВК) - ведется протокол, (ПС) - протокол не ведется, (ВШ) - восстанавливается протокол.

- П Р И М Е Ч А Н И Я
- Символы РУС и ЛАТ (восьмичисловые коды 16 и 17) вводятся в буфер и файл, но на экране не отображаются. Их наличие и местоположение можно определить по задержке перевода движения курсора в строке.
 - Символ перевода формата отображается на экране как <FF> или <ff>.
 - Тип редактируемого файла по умолчанию .PAS;
Имя файла сохранения по умолчанию SAVE.TMP;
Имя макрофайла по умолчанию К7401.MAK;
Тип файла протокола редактирования .JOU (размер 10 блоков).
 - Справочный файл К7401.HLP должен находиться на системном диске 5V:.
 - Уровень вложения макрокоманд до 20.
 - Длина (ТЕКСТ) при поиске не более 17 символов, включая РУС и ЛАТ.
 - Команды восстановления последнего удаления <0> можно использовать после команд <2>, <4>, <5>, <6> для переноса строки или ее части.
 - Редакторы SED вводят входной файл частями либо до первого символа <FF>, либо до заполнения буфера (150 строк). Для продолжения редактирования необходимо воспользоваться командами ПФ1 I или ПФ1 N (ВВОД).
 - При работе редакторов SED дополнительная цифровая клавиатура генерирует только команды. Для восстановления командного режима дополнительной цифровой клавиатуры после сбоя в работе дисплея необходимо выполнить команду редактора ОБНОВЛЕНИЕ ЭКРАНА.

ный текст, комментарии по использованию, командные файлы генерации и запуска, загрузочные модули для различных типов процессоров, могут

быть поставлены на ГМД диаметром 203 мм в формате РАФОС.

Телефон 441-11-77, Киев

Сообщение поступило 30.11.87

* Операционная система СМ ЭВМ РАФОС: Справочник / Под общ. ред. В. П. Семика. - М.: Финансы и статистика, 1984. - 207 с.

Е. И. Волкова, П. А. Юнусов

УНИВЕРСАЛЬНЫЙ ЭКРАННЫЙ РЕДАКТОР ОБЩЕГО НАЗНАЧЕНИЯ DREAM ДЛЯ БЕЗДИСКОВЫХ СИСТЕМ НА БАЗЕ МИРКОЭВМ «ЭЛЕКТРОНИКА 60»

Дисковые конфигурации микроЭВМ «Электроника 60М» поддерживаются поставляемой в системе комплекса операционной системой (ОС) ФОДОС и генерируемой для этих комплексов ОС РАФОС, которые обеспечивают широкий круг применений. Они имеют, как правило, универсальные экранные редакторы, предоставляющие оператору редактировать любую строку текстового файла с помощью функциональной клавиатуры, видя перед собой часть текста.

Бездисковые конфигурации микроЭВМ «Электроника 60М» поставляются с перфоленточной ОС (ПЛОС), куда входит редактор текста, реализованный для комплекса с печатающим устройством CONSUL-260. При появлении дисплейных терминалов с микропроцессорным управлением, обладающих повышенными возможностями работы в автоматическом режиме, использование упомянутого выше редактора стало нецелесообразным для работы с дисплеями типа 15ИЭ-00—013, VDT-52129, VDT-52130 из-за отсутствия экранного режима.

Предлагаемый редактор DREAM предназначен для создания новых и корректирования старых текстовых файлов. Он выполнен на базе редакторов дисковых ОС (K200, SED, TED, USED).

Экранный редактор имеет три режима работы: управляющий, экранный, командный.

В управляющем режиме пользователь может получить справочную информацию о наличии свободной области памяти, произвести операции по очистке текстового буфера, продвижению по буферу, обслуживанию стандартных внешних устройств (фотосчитывателя, терминала, перфоратора, АЦПУ).

В экранном режиме ввод и редактирование текста, а также передвижение маркера по экрану выполняются с помощью алфавитно-цифровой клавиатуры и специально предназначенных для этого функциональных клавиш дисплея:

←	передвигает маркер на одну позицию влево;
→	передвигает маркер на одну позицию вправо;
↑	перемещает маркер на одну строку вверх в ту же позицию. Если маркер находился в первой строке экрана, то перемещение не происходит;
↓	перемещает маркер на одну строку вниз в ту же позицию. Если маркер стоит в последней строке экрана, то перемещение не происходит;
↖	перемещение маркера в начало страницы текста, отображенной на экране;
↘	перемещение маркера в начало следующей строки;
↙	Перемещение маркера в крайнюю правую позицию последней строки текста (в конец страницы);
ВК	перемещение маркера в начало текущей строки. Аналогично ↖;
ГТ	перемещение маркера вправо до следующей позиции табуляции ряда: 9, 17, 25, 33, 41, 49, 57, 65, 73, 74, 75, 76, 77, 78, 79;
ЗБ	стирание символа, непосредственно предшествующего маркеру, и перемещение влево маркера на одну позицию;
→	сдвиг текста на шаг вправо по строке от позиции маркера (размыкание);
←	сдвиг текста на шаг влево по строке (смыкание);
—	образует пустую строку над строкой, указываемой маркером (сдвиг текста вниз);
↓	удаляет строку, указываемую маркером. Место удаленной строки занимает строка, стоявшая под удаляемой — если она есть (сдвиг текста вверх).

В предлагаемой версии в отличие от упомянутых редакторов отредактированный текст запоминается путем чтения данных из ОЗУ дисплея с фильтрацией и записи их в более компактной форме в текстовый буфер ОЗУ ЭВМ.

Редактор DREAM предоставляет возможность пользователю поиска цепочки символов. Нажатие клавиши AP2 вызывает отображение в первой строке экрана приглашения к заданию аргумента поиска «Что поищем?», который редактор будет пытаться отыскать в текстовом буфере. Если эта попытка окажется успешной, то на экране отобразится страница буфера, на которой встретилось первое появление заданной цепочки символов. Поиск начнется с текущей страницы, и маркер отметит на экране найденный аргумент поиска. В случае безуспешного поиска на экране отобразится сообщение «Текст не найден».

С помощью команд редактора DREAM можно сделать следующее: стереть, вставить или продублировать N символов; удалить, переместить или размножить N строк; «разрезать» или «склеить» две соседние строки; вывести на экран строку, на которую указывает курсор; вывести на экран строки с их порядковыми номерами.

Работа в трех режимах редактора DREAM позволяет более осмысленно использовать управляющие клавиши терминала и не производить их многократного нажатия для выполнения какой-либо операции. Предлагаемый редактор прост и удобен в эксплуатации, не требует высокой квалификации пользователя.

Минимальный объем требуемой оперативной памяти для эксплуатации редактора DREAM 8 Кбайт.
Телефон 64-51-69, Воронеж

ЛИТЕРАТУРА

1. ЭВМ «Электроника 60». Техописание. Кн. 2.
2. ЭВМ «Электроника 60». Перфоленточная операционная система. Программное обеспечение. Кн. 6.
3. Программное обеспечение СМ ЭВМ. ОС РАФОС 11. Редактор общего назначения.
4. Программное обеспечение СМ ЭВМ. Универсальный экранный редактор USED.

Статья поступила 20.02.87

УДК 681.03.06

Ю. Б. Уткин

БАЗОВЫЙ РЕДАКТОР СИМВОЛЬНОЙ ИНФОРМАЦИИ ESO

Программа ESO — упрощенный вариант редактора символической информации. Она предназначена для формирования текста в памяти ЭВМ и на магнитном носителе. Алгоритм работы построен по принципу «Ромашки»; центральное ядро программы реализует перечень зарезервированных директив, позволяющих выполнить базовые функции по первоначальному формированию и последующему ведению массивов данных произвольной структуры. После их выполнения программа вновь возвращается в свое ядро.

Программа написана на языке БЕЙСИК версии 02 и работает на микроЭВМ «Искра 226». Объем используемой памяти: программа — 13,0 Кбайт; входные и выходные данные по 16,0 Кбайт; прочие переменные — 2,2 Кбайт; свободная область — 17,4 Кбайт.

Программу можно легко модифицировать под любую версию языка БЕЙСИК для любой ЭВМ.

Принцип работы программы построен на наличии в памяти ЭВМ двух крупных символических массивов данных — входного и выходного. Над ними могут быть выполнены идентичные по своему функциональному назначению операции — директивы (некоторые могут реализовать связь массивов).

В программе ESO сформированы входной и выходной массивы, каждый на 200 строк по 78 символов в строке (10 страниц), которые можно расширять и су-

жать как по длине, так и по ширине строки. Число страниц входного массива может быть увеличено до 30 (600 строк, 48 Кбайт). При этом выходной массив должен содержать не более одной страницы (отсутствует место для расширения программы в ОЗУ ЭВМ).

Оба массива могут быть записаны на магнитном диске каталога файлов: входной — под именем XXXX.ESO, выходной — под именем XXXX: ESO, где XXXX — произвольный набор символов.

На экране дисплея постоянно отображены входные или выходные данные, а для реквизитов и режимов работы зарезервированы специальные строки экрана дисплея. Такое построение программы повышает активность пользователя при работе с данными и предоставляет ему минимально необходимую информацию о способах управления этими данными.

Структура экрана дисплея по строкам: 1 — реквизиты программы, данных; 2 и 23 — структура данных, 3 ... 22 — сами данные; 24 — перечень зарезервированных директив.

Взаимодействие оператора с программой основано на принципе непосредственного редактирования информации*. Полностью отказаться от меню (явного или скрытого в инструкциях оператора) невозможно, но его можно свести к минимуму. Это реализовано при помощи строки-подсказки, которая постоянно присутствует на экране дисплея, занимает минимум места и содержит перечисление всех используемых в программе директив. При нажатии зарезервированной клавиши выбранная директива вводится в программу. Предполагается, что оператор должен быть предварительно хорошо подготовлен для работы с ESO. Для этого в системе имеются файлы инструкций. Операции над входным и выходным массивами данных выполняются с помощью следующих директив: <DIM> — задание размеров массива, <CLEAR> — очистка массива, <CR/LF> — вход в строку для ее редактирования (после этого осуществляются стандартные директивы — горизонтальное перемещение курсора влево и вправо, возврат на шаг, набор, вставка, удаление символа, стирание остатка строки, переводы старой строки), <INSERT> — вставка строки, <DELETE> — удаление строки, <LIST> — листание страницы, <PRINT> — печать страницы, <SELECT> — назначение дискового устройства, <LOAD> — загрузка данных с диска, <SAVE> — запись данных на диск, <RUN> — допрограммируемая директива, позволяющая реализовать связь входного и выходного массивов. Кроме того, имеются директивы перемещения специального курсора вверх или вниз по экрану для указания строки.

Таким образом, ESO — базовая программа для широкого круга прикладных задач. Ее возможности:

использование произвольной внутренней структуры входного и выходного массивов при формировании данных (например, неупорядоченных текстовых массивов, буквенно-цифровых таблиц);

программное расширение или сужение входного-выходного массивов по длине и (или) ширине;

допрограммирование директивы <RUN> под свои конкретные цели;

ввод новых, не предусмотренных ранее директив (например, ввод директив работы с файлами — создание, уничтожение, переименование, чтение каталога файлов; расширение набора директив редактирования — перенос, растяжка и запоминание строки, табуляция, поиск по контексту, автоматическая нумерация строк; ввод директив работы с табличными базами данных — копирование частей массива, его проверка на соответствие структуре, сортировка по заданному образцу; ввод ди-

ректив математических операций над столбцами и строками таблиц — суммирование столбцов и строк в бухгалтерских расчетах);

модификация программы для любой ЭВМ.

Телефон 57-86-13, Рига

Статья поступила 04.01.88

Письмо в редакцию

УВАЖАЕМАЯ РЕДАКЦИЯ!

Написать письмо Вам меня побудила статья Е. С. Тимофеева, В. Н. Васильева, Н. П. Васильева «Организация сложных файловых систем в среде ОС РАФОС» в «МП» № 2, 1988 г. Моя работа напрямую связана с ОС РАФОС (системное программирование РАФОС II и ДОС), поэтому изложу некоторые свои мысли.

1. Вероятнее всего, авторы работают в ОС РАФОС II ПЛЮС. Вызывает удивление затраченный труд на организацию «многопользовательского режима работы, динамическое распределение памяти». У нас работают на четырех рабочих местах 14 программистов, у каждого своя задача, и никаких проблем с иерархией. Но вот работает ли в РАФОС II RT-11, монитор TSX-6 Plus, который позволяет организовать двухуровневую систему виртуальных дисков, что, на мой взгляд, вполне достаточно?

2. По поводу «защиты от несанкционированного доступа». Если задаться целью уничтожить какую-то информацию, то это вряд ли спасет (от квалифицированного программиста). Ну, а от неподготовленных пользователей или прикладных программистов такую же, если не лучше (с более широким выбором привилегий), обеспечивает программа 2060U.

3. «Изменение размера области» также не устраивает. Если пользователю необходимо дополнительное пространство на носителе, то для него создается еще один виртуальный диск.

В заключение хочется сказать вот о чем. К сожалению, носители типа CM5400, ES5061 нет-нет да и искажают файловую структуру. Особенно «печальны» последствия при «зависании» ЭВМ при «сжатии» 29-Мбайт пакетов. Системные программы, существующие в ОС РАФОС II, предназначены для коррекции, ориентированы на файлы. Причем программа SIPP не выполняет корректировку, если есть стойкие блоки в каталоге (PATCH на это не обращает внимания). Мною разработаны три программы:

1) поиска SEA (ищет каталоги логических дисков или любую заданную информацию по образцу, размер 8 Кбайт);

2) проверки каталоговой структуры (проверяет соответствие длины области файлов каждого сегмента длине данного сегмента (всего каталога), размер 8 Кбайт);

3) корректировки каталога КОР (позволяет просмотреть любой заданный сегмент каталога, в том числе по отдельности 0-й и 1-й блоки сегмента, в формате файловой структуры, причем при распечатке можно сразу же произвести коррекцию, не задавая формата: он устанавливается программно, размер 8 Кбайт).

Системным программистом из ВЦ СО АН СССР Калантаевым П. А. разработан высокоэффективный экраный редактор TNE, занимаемый объем — 37 Кбайт, который адаптируется (настраивается) к любому термину, используемому в системе (BTA 2003, 2000—15, 2000—30, VIDEOTON, MERA, Электроника 15 ИЭ).

С помощью данных программ был неоднократно восстановлен 29-Мбайт пакет с разрушенной при сжатии каталоговой структурой.

142400, Московская обл., Ногинск, ул. Комсомольская, д. 24, кв. 21, Кириченко Сергей Григорьевич
(Ответ на это письмо см. на с. 67)

* Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации. // Микропроцессорные средства и системы, — 1986. — № 1.

П. В. Вельтмандер, С. И. Власов

ОФОРИС — ПРИКЛАДНАЯ ГРАФИЧЕСКАЯ СИСТЕМА ФОРМИРОВАНИЯ ИЛЛЮСТРАТИВНЫХ МАТЕРИАЛОВ

Прикладная графическая система ОФОРИС разрабатывалась в качестве средства, позволяющего оперативно готовить, редактировать и документировать различные иллюстративные материалы, такие как плакаты, мультипликации, заставки, рекламные проспекты и т.п., задавать сценарий показа заготовленных изображений и собственно организовывать показ. Изображение строится в диалоговом режиме за пультом дисплейной станции ГАММА-4.2 [1] и состоит из примитивных плоских и псевдообъемных элементов, которые могут накапливаться и сохраняться во внешней памяти в именованных сегментах. Суммарное изображение на экране дисплея формируется из изображений вызванных сегментов и отдельных примитивов, не входящих в сегменты.

Построенные изображения (см. вкладку на с. 1, 3) могут подвергаться геометрическим и цветовым преобразованиям. При этом изменяются как сегменты целиком, так и отдельные составляющие их примитивы. Предусмотрены геометрические преобразования перемещения, масштабирования и поворота. Цветовое преобразование заключается в смысле цвета примитива на требуемый. Предоставлены средства стирания отдельных примитивов, и (или) сегментов, и всего изображения целиком. У ломаных можно стирать отдельные звенья.

Сценарий показа формируется заданием требуемой последовательности вызовов сегментов на экран и индивидуальным установлением (опросом, заменой) времени высвечивания сегмента на экране.

Полученное изображение может документироваться в виде копии экрана средствами базового программного обеспечения ГАММА-4.2 [2]. Кроме этого, для подготовки иллюстраций в виде плакатов изображение, состоящее из вызовов сегментов, может быть интерпретировано в файл для графопостроителя ЭМ-7042А или БЕНСОН-222. В этом случае вычерчиваются контуры примитивных элементов. При задании специального режима контуры заштриховываются в соответствии с их цветами и яркостями, которые выбираются экспериментально.

Входной язык

Входной язык системы ОФОРИС — директивного типа. Каждая директива состоит из ее идентификатора и списка параметров, который может быть пустым. Идентификатор директивы задается либо указанием ее имени на меню директив с помощью координатного устройства, либо выбором требуемой пиктограммы на преддисплейном планшете, либо вводом ее номера с алфавитно-цифровой клавиатуры. Для задания некоторых, наиболее часто употребляемых директив используется функциональная клавиатура. Появление меню на экране может быть автоматическим (после исполнения очередной директивы) или вызываться нажатием на функциональную клавишу. Параметрами директив являются числа, координаты, имена, произвольные строчки алфавитно-цифровых символов, а также коды клавиш функциональной клавиатуры.

По функциональному признаку директивы разбиты на пять групп: управления, построения примитивов, задания атрибутов примитивов, преобразования изображений, дополнительные.

Ряд директив завершается после однократного выполнения задаваемых ими действий, например директивы управления. Многие директивы предполагают повторение выполняемой ими работы до нажатия функциональной клавиши выхода из директивы. К ним относятся директивы построения, задания атрибутов и преобразований.

Следует отметить, что, хотя директивы все одного уровня иерархии, тем не менее в процессе выполнения директив построения можно, не прерывая их исполнения, переустановить атрибут цвета.

Директивы управления:

включить/выключить базу данных (разрешение/запрещение запоминания сегментов строящегося изображения в файлах на внешней памяти);

открыть сегмент (открытие на запись сегмента, в котором будут запоминаться дальнейшие построения). В качестве открываемого сегмента можно использовать вновь создаваемый сегмент и сегмент, уже имеющийся в базе данных;

перевыдать изображение (перевыдаются сегменты из списка вызванных на экран). Изображения примитивов, находящиеся вне сегментов, стираются. Если сегменты создавались с ненулевым временем их показа, то эта директива может обеспечить циклическую перевыдачу подготовленного сценария заданное число раз;

объединить сегменты (слияние всех сегментов, вызванных на экран, в единый сегмент с именем текущего сегмента). Объединяемые сегменты могут либо сохраниться в базе данных, либо быть уничтожены;

конец работы (закрываются все текущие файлы и заканчивается работа программы).

Директивы построения изображения. Геометрическая информация задается в растрах. Начало координат — точка (0, 0) в левом нижнем углу. Максимальное значение координаты — 2047 дискрет. Наиболее удобен ввод координат с помощью шара (трекбола); но возможен и ввод чисел с клавиатуры.

В пределах исполнения одной директивы может быть выполнена совокупность однотипных построений, например после выбора директивы построения ломаной строится не одна, а группа ломаных. Для указания о завершении очередного построения в группе используется определенная функциональная клавиша. Для части примитивов с фиксированным числом данных, требующихся для одного построения, нажатие этой клавиши необязательно. Выход из исполнителя директивы обеспечивается нажатием на одну и ту же функциональную клавишу для всех директив.

Внешний вид построений определяется текущими значениями соответствующих атрибутов примитивов, задаваемых с помощью директив установления атрибутов. Предоставлена возможность установления атрибута цвета не только специальной директивой, но и нажатием на определенную функциональную клавишу во время исполнения директив. Например, при построении группы ломаных их можно выполнять разного цвета.

Группа директив построения:

ломаных. Вводимые координаты вершин ломаной соединяются отрезками с текущими типом линии и цветом построения. Возможна последовательная отмена построенных отрезков;

многоугольников. Координаты вершин многоугольника задаются так же, как и для ломаной. После нажатия функциональной клавиши завершения ввода отдельного многоугольника полученный многоугольник заливается текущим цветом;

строк текста. Задаются координаты начала строки, затем с клавиатуры вводится строка символов, отображаемая в форме, определяемой атрибутами строки;

дуг окружностей. Дуга определяется заданием координат трех ее точек или вводом с клавиатуры радиуса, углов начальной и конечной точек дуги и заданием координат центра;

кругов. Круг определяется заданием либо координат центра и вводом радиуса, либо координат центра и точки на окружности, либо координат трех точек на границе;

колец. Кольцо определяется заданием координат центра и вводом значений минимального и максимального радиусов или заданием координат центра и точек на внутренней и внешней окружностях кольца;

заполненных эллипсов. Эллипс определяется заданием координат центра и вводом размеров полуосей и угла наклона большой полуоси относительно направления оси X или заданием координат точки пересечения большой полуоси с границей, координат пересечения малой полуоси с границей и координат одного из фокусов;

псевдообъемных изображений. Предусмотрено рисование псевдообъемных фигур кистями пяти типов и построение псевдообъемных тел — шара, тора и эллипсоида. Эффект псевдообъемности обеспечивается неравномерной интенсивностью окраски. Тип и размеры кисти, тип псевдообъемного тела и правила подсветки псевдообъемных изображений задаются установлением соответствующих атрибутов. Рисование кистями типов 0—2 с равномерной закраской, повышенной яркостью сбоку (подсветка сбоку) и в центре (подсветка прямо) заключается в формировании следа за перемещением перекрестия. Рисование кистями типов 3—4 аналогично построению ломаной. Отличие заключается в том, что вершины ломаной соединяются «толстыми» линиями с подсветкой сбоку или прямо. Способ ввода информации, определяющей построение шара, тора и эллипсоида, совпадает с таковым для круга, кольца и эллипса соответственно. Для шара требуется дополнительно указать точку расположения источника освещения данного шара. Для тора и эллипсоида максимум интенсивности располагается в середине.

Директивы установления атрибутов:

типа ломаной (непрерывная, штриховая, штрихпунктирная);

номера размера символа (1 — матрица 7×9 точек, 2 — 14×18 и т. п.);

ориентации вертикали литерной площадки символа вдоль одного из четырех направлений осей координат; направления вычерчивания строки относительно вектора вертикали символа (вправо, влево, вверх, вниз); типа псевдообъемной фигуры и номера кисти (плоская кисть с равномерной подсветкой, объемная кисть с подсветкой сбоку и прямо, тор с прямой подсветкой, шар с изменяемой подсветкой, эллипсоид с подсветкой прямо);

задание номера ширины кисти;

цвета построений с помощью указания на цветовую палитру или точку изображения, имеющую пухлый цвет; если выбор цвета из-за недостаточного размера палитры либо изображения затруднен, то с помощью аппаратного трансфокатора дисплея можно увеличить масштаб и более точно задать позицию с пухлым цветом;

цвета фона символов, аналогична выбору цвета построений.

Директивы преобразований:

геометрические преобразования указанного сегмента (перемещение в заданную позицию, поворот по и против часовой стрелки вокруг осей координат, уменьшение и увеличение сегмента);

удаление указанного сегмента;

геометрические преобразования указанного примитива (перемещение, масштабирование, повороты по и против часовой стрелки вокруг осей X и Y);

удаление указанного примитива;

смена цвета указанного примитива.

Указание выполняется помещением перекрестия на требуемый объект, возможно, при этом для более точного указания требуется использование аппаратных трансфокаций и протяжки изображения. После нажатия функциональной клавиши, отмечающей выполнение указания, система начинает мерцанием выделять первый из объектов, попавших в зону захвата перекре-

стия. Правильность указания подтверждается нажатием функциональной клавиши. Если в зону захвата перекрестия попало несколько объектов, то последовательными нажатиями на соответствующую клавишу можно вызывать мерцание объектов. По завершении указания производится собственно преобразование.

Для перемещения необходимо задать новую позицию. Изображение указанного объекта перемещается так, что в новую позицию переместится точка объекта, на которую выполнялось указание.

Выбор требуемых преобразований масштабирований и поворотов (уменьшение, увеличение, поворот по/против часовой стрелки) осуществляется с помощью функциональной клавиатуры. Выбранное преобразование многократно выполняется с определенным шагом в течение времени удержания клавиши относительно точки объекта, использованной при выполнении указания. В обычном режиме исходное изображение убирается с экрана и замещается на преобразованное. В ряде случаев полезно оставить оба изображения. Требуемый режим выбирается функциональной клавишей подтверждения указания.

Преобразование завершается по нажатии соответствующей функциональной клавиши. В зависимости от ее выбора либо восстанавливается исходное состояние изображения, либо фиксируется результат преобразования и изображение обновляется. При обновлении с экрана будут убраны все промежуточные фазы изображения, если преобразование выполнялось с оставлением промежуточных состояний на экране.

Дополнительные директивы, повышающие удобства работы пользователей:

выдать либо имя текущего сегмента, либо имена выделенных сегментов, либо имена всех сегментов базы данных;

включить/выключить эхо трекбола либо в виде резиновой нити от старой к текущей позиции, либо в виде непрерывного следа его перемещения;

включить/выключить вспомогательный позиционирующий растр, имеющий шаг в 10 растров;

установить шаги для аппаратных операций «протяжка» и «трансфокация» дисплея, шаги масштабирования и поворота для геометрических преобразований изображения, время высвечивания очередного вызванного сегмента.

Реализация

Система ОФОРИС представляет собой задачу, развитую на функционально-ориентированные сегменты (управление, построение, установление атрибутов и т. д.). Программы написаны на языке Фортран и функционируют в среде ОС РАФОС с SJ-монитором. Суммарный объем программного кода — около 4000 строк. Из-за чрезвычайно жестких ограничений на память (под программу имеется всего около 40 Кбайт) оказалось невозможным использование средства манипулирования иерархическими структурами данных. Все сегменты базы данных представлены в виде файлов последовательного доступа ОС РАФОС на электронном диске. Имена вызванных сегментов хранятся в специальном списке. Для сохранения сеанса работы, при возникновении опасности нехватки места под оперативные файлы и данные, текущие файлы закрываются, программа переходит в состояние «стоп». После уплотнения файлов и вызова задачи работа может быть продолжена.

Заключение

Разработка системы ОФОРИС проводилась на основании опыта, накопленного при эксплуатации системы видеографики [3]. Время обучения работе с системой — 1—2 сеанса для пользователя, имеющего опыт работы с вычислительной техникой, и 1—2 дня — для художников-операторов. В настоящее время ведутся работы по включению в состав системы средств построения фрактальных поверхностей (в несколько упрощенной постановке, рассчитанной на автономную вычислительную

мощность дисплейной станции). Наличие средств связи с главными ЭВМ типов БЭСМ-6, ЕС-ЭВМ и мультипроцессорами ПС-2000 позволит решать задачи построения фрактальных поверхностей и иные задачи видеографики, требующие большого объема вычислений.

Предполагается расширение системы средствами автоматической подсказки, переустановления ее параметров (система координат, зона захвата перекрестия, шаги позиционирующего раstra и т. п.), генератором художественных шрифтов, задания области на экране и сохранения попавшего в нее изображения в виде сегмента, т. е. создания геометрически упорядоченных сегментов в отличие от текстуально упорядоченных. Расширение ограничено ресурсом оперативной памяти.

Телефон 32-38-39, Новосибирск

ЛИТЕРАТУРА

1. Сизых В. Г. Растровые дисплеи ряда ГАММА.— Препринт.— Новосибирск, 1985. (ВЦ СО АН СССР, № 607.)
2. Бучнев А. А., Сизых В. Г. Базовое программное обеспечение дисплея ГАММА-4.2 // Тез. докл. II Всесоюз. конф. «Методы и средства обработки сложной графической информации».— Горький, сент. 1985.
3. Бучнев А. А., Сизых В. Г. Система видеографики на базе дисплейной станции ГАММА-4.2 // Тез. докл. Всесоюз. школы-семинара «Информатика и интерактивная компьютерная графика».— Цахкадзор, март 1987.— С. 24—25.

Статья поступила 05.05.88

УДК 681.3.06

С. К. Дюкарев

ГРАФИЧЕСКИЙ ПАКЕТ «КАЛКОМП» ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Широкое распространение получил пакет программ фирмы КАЛКОМП* построения графической информа-

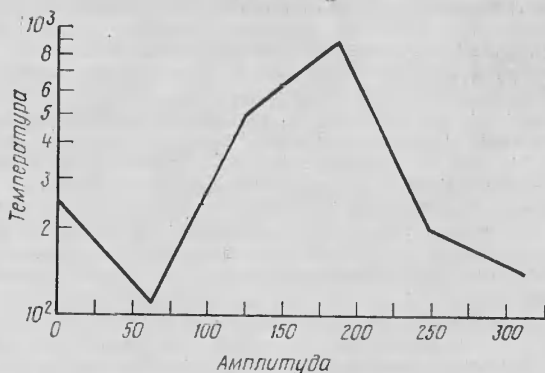
* Calcomp graphics functional software, USAS FORTRAN/GENERAL, July 1975.

ТАБЛИЦА СИМВОЛОВ ПОДПРОГРАММЫ		SYMВ	
ПЛОТТЕР 'АТЛАС - 5' ДЛЯ ЭВМ ТИПА СМ			
0	□ 15	32	48
1	○ 17	33	49
2	△ 18	34	50
3	+ 19	35	51
4	× 20	36	52
5	◇ 21	37	53
6	⊕ 22	38	54
7	⊗ 23	39	55
8	⊚ 24	40	56
9	⊙ 25	41	57
10	⊛ 26	42	58
11	⊜ 27	43	59
12	⊝ 28	44	60
13	29	45	61
14	⊗ 30	46	62
15	- 31	47	63

ЦЕЛОЕ ПРИ ВЫВОДЕ СИМВ ПОКАЗЫВАЕТ СПРАВА ЕГО НАЧЕРТАНИЕ

Рис. 1

Подпрограммы SCALG, LGAXS, и LGLIN



Подпрограмма POLAR

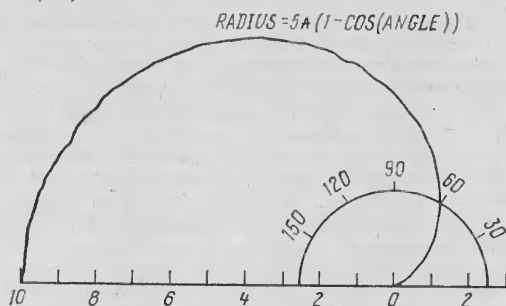


Рис. 2

ции на векторных плоттерах. Фирма поставляет отдельные комплекты программ, подпрограмм, включая тексты, и сопровождает их необходимой документацией. Имеется опыт адаптирования этого программного обеспечения для ЭВМ типа «Электроника 60» под управлением операционной системы РАФОС. Построенный по иерархическому принципу графический пакет разбивается на три уровня. Верхний уровень содержит программы построения электронных схем, общих и экономических графиков, вычерчивания блок-схем, географических карт, трехмерных изображений. Средний уровень — подпрограммы построения часто используемых изображений: планово-экономических данных, кривых, чертежей и различных графиков. Нижний уровень — основа пакета, подпрограммы генерирующие управляющие последовательности для используемого плоттера. Пакет содержит девять подпрограмм на Фортране и одну на Макроассемблере, которая обеспечивает настройку на заданный плоттер.

Объем библиотеки нижнего уровня около 140 блоков. Значительно была переработана подпрограмма вывода символов. Символы полностью формируются программно. Не используется аппаратура плоттеров, генерирующая символы микропрограммно. Заменены прописные латинские буквы на русские (рис. 1).

Для вывода изображений (рис. 2) использован плоттер «Атлас-5», подключаемый либо к шине микроЭВМ, либо к НМЛ или перфоленте. Для записи изображений на магнитную ленту использован востроенный драйвер, что позволяет адаптировать пакет к любым используемым НМЛ.

Телефон 135-84-16, Москва

Сообщение поступило 18.04.88

ИНТЕРАКТИВНАЯ ГРАФИЧЕСКАЯ СИСТЕМА ГРАФ

Интерактивная графическая система ГРАФ предназначена для создания в режиме диалога принципиальных электрических схем, демонстрационных чертежей и рисунков двумерных объектов, а также их хранения, редактирования и вывода на принтеры различных типов.

Система ГРАФ разрабатывалась как одно из средств повышения уровня автоматизации проектирования и документирования программных и технических систем. Она реализована на 32-разрядной ЭВМ Labtam-3215 в ОС Unix на языке Поляр [1, 2], что позволило, с одной стороны, проверить возможности как самого языка, так и транслятора с этого языка, а с другой — при создании системы ГРАФ не заботиться о поддержке динамической структуры рисунка и вводе-выводе сложных структур данных, а целиком сосредоточить свое внимание на разработке интерфейса, алгоритмов и технологии редактирования. Один из основных принципов, которому старались следовать разработчики, — это простота общения с системой и ее надежность.

Проанализировав существующие формы человеко-машинного взаимодействия, авторы пришли к выводу, что наиболее адекватным решаемой задаче будет интеллектуальный интерфейс [3], т. е. некоторая среда, отделяющая пользователя от системного программного обеспечения. Немалое значение имеет возможность адаптации этой среды к конкретному пользователю.

Основные компоненты системы ГРАФ: монитор с экраным интерфейсом; экраный графический редактор; программы создания копий рисунков на бумаге; программы настройки графической системы. Полный объем системы — около 450 Кбайт, в том числе монитор — 77 Кбайт, редактор — 130 Кбайт.

В существующей реализации в качестве графического терминала используется рабочая станция Labtam-3000 вычислительного комплекса Labtam-3215. Графический язык — REGIS (стандарт фирмы Digital Equipment Corporation), входящий в состав программного обеспечения рабочей станции.

Экраный интерфейс

При создании системы определился единый стиль экранного интерфейса для программ системы и единая технология работы с объектами — файлами в мониторе и графическими элементами в редакторе. На экране открыто несколько окон: рабочее меню, для вывода служебной информации и подсказки. Расположение окон фиксировано, так как работа в системе ведется на удаленных терминальных станциях, и обновление экрана занимает достаточно длительное время.

Интерфейс основан на непроектурном объектно-ориентированном взаимодействии: пользователь сначала выбирает объект обработки, а затем — необходимую команду из предлагаемого списка команд (меню). Меню содержит только те команды, которые допустимы при выполнении связанных с объектом действий. Тем самым достигается защита от несанкционированных действий и обзорность набора допустимых команд. Состав меню динамически изменяется в зависимости от складывающейся ситуации и помогает пользователю определить, какой этап работы выполняется.

В системе можно использовать небольшое количество управляющих клавиш для передвижения по меню и в рабочем окне, выполнения выбранной команды. Настройка системы на конкретные клавиши может быть проведена пользователем из монитора системы, в даль-

нейшем эта информация хранится в специальном файле инициализации.

Формирование меню и выбор команд

Оператор ввода данных в программе всегда выполняется в некотором контексте, к которому можно отнести положение этого оператора в программе, текущее состояние переменных и т. д. В редакторе такой контекст составляют текущие координаты рисунка, сегмент, редактируемый примитив, состояние буфера, значения параметров редактирования, в мониторе — рабочий архив, текущий рисунок. Анализируя контекст в момент ввода информации, можно определить состав команд, допустимых в этом контексте, сформировать и предложить пользователю соответствующее меню. Такой метод смены меню реализован в системе.

Полный набор команд хранится в служебном файле. Формат команды: (код команды, имя команды, комментарий). Файл меню можно модифицировать: изменять имена команд или их порядок, удалять часть команд.

Выбирать команду в предложенном меню можно двумя способами:

- с помощью клавиш передвижения по меню;
- набором имени команды. Имя не обязательно вводить полностью; поиск команды производится сравнением введенной строки с начальными символами имени каждой команды из текущего меню. Выбирается первое имя, в котором префикс совпал с введенной цепочкой символов.

Исполняются команды при нажатии клавиши **выполнить команду**.

Приведенный механизм работы с меню используется и для других целей. Например, при работе редактора с архивом рисунков (чтение рисунка в буфер редактора или запись содержимого буфера в архив). Каталог файлов, доступных для чтения (записи) выводится в виде меню. Имя нужного файла выбирается описанным способом. Если меню превышает размеры окна, то оно выводится порциями. Новая порция выдается, если нужное имя или команда находится за границами окна.

Во время переключения по меню можно в любой момент вызвать подпрограмму подсказки, выдающую комментарий к команде.

Монитор системы графического редактирования ГРАФ

На входе в систему ГРАФ начинает работать программа, называемая монитором, предоставляющая пользователю все возможности системы. Из монитора доступны подпрограммы: вывода рисунка на устройство печати, настройки монитора и редактора, экранного редактирования рисунков.

Монитор позволяет выполнять весь набор необходимых действий по ведению архива рисунков и управлению прикладными подпрограммами системы ГРАФ, обеспечивая надежную работу нескольких пользователей с объектами одного архива. Перед выполнением любого действия, приводящего к необратимой модификации или уничтожению объекта, предлагается подтвердить поданную команду.

Интерфейс организован таким образом, что требует от пользователя минимальных действий. На экране имеются окна для каталога, меню и служебной информации. Окно меню расположено в левой стороне экрана, в нем в одну колонку выводятся команды текущего меню. Окно для вывода запросов и подсказки находится в верхней строке экрана. Остальную часть экрана занимает рабочее окно, в котором выводится каталог с именами рисунков личного архива пользователя.

В каждый момент работы существуют текущие имя рисунка и команда, которые выделяются специальным образом.

Распечатка рисунков

Программы распечатки рисунков служат для вывода рисунков на матричные графические принтеры в заданном масштабе. Система настраивается на тип принтера и размеры бумаги через специальный файл, в котором должна содержаться необходимая информация: формат вызова программ печати и параметры печати.

Если рисунок не помещается на листке бумаги, он разбивается на меньшие части (кадры) и выведен на нескольких листах. Порядок вывода кадров — сверху вниз и слева направо. Есть возможность задать автоматический выбор масштаба, при котором рисунок размещается на одном листе бумаги. Через параметры можно определить, какого цвета фигуры не выводить при прорисовке рисунка.

Настройка графической системы

Программы настройки позволяют:

поставить в соответствие командам монитора и редактора подходящие для данного пользователя управляющие клавиши, отличные от установленных по умолчанию;

установить более удобные начальные параметры графического редактора;

редактировать шрифт (изменить начертание букв, создать свои символы).

Экранный редактор рисунков

Графический редактор позволяет создавать и модифицировать рисунок на экране дисплея, используя управляющие клавиши. При этом он оперирует с изображением рисунка, за которым стоит динамическая структура данных, написанная на языке Поляр, объединенных рекурсивным типом:

рисунок-точка ^ (цвет: ЦЕЛ, точ: коорд),

отрезок ^ (цвет: ЦЕЛ, верш1, верш2: коорд),

прямоугольник ^ (цвет: ЦЕЛ, верш1, верш2: коорд),

линия ^ (цвет: ЦЕЛ, верш: <коорд>),

многоугольник ^ (цвет: ЦЕЛ, верш: <коорд>),

окружность ^ (цвет: ЦЕЛ, точокр, центр: коорд),

текст ^ (цвет, угол, размер, отражение: ЦЕЛ, тп: коорд, текст: <ЛИТ>),

сегмент ^ (рисунок),

где коорд — координаты в абсолютной системе координат, а описание <тип> определяет последовательность значений данного типа; длина этой последовательности не фиксирована и может динамически меняться. Новые элементы рисунка создаются из элементарных геометрических фигур — примитивов (точка, отрезок, прямоугольник, ломаная линия, многоугольник, окружность и текст). Перечисленные элементарные фигуры входят в описанный выше тип в качестве вариантов.

Отдельные примитивы могут быть объединены в сегменты. Так образуется древовидная структура рисунка. Объединение происходит либо неявно с помощью некоторых операций, либо явно — соответствующей командой. Над сегментами можно выполнять те же операции, что и над примитивами. При этом операция осуществляется над каждым примитивом, входящим в сегмент.

Пользователю предоставляется возможность выбора одного из двух режимов работы с сегментами: при первом режиме объединение примитивов в сегменты никак не отражается на изображении рисунка на экране, при втором — примитивы, принадлежащие редактируемому сегменту, выделяются цветом, что позволяет ориентироваться в структуре рисунка.

В начале работы рисунок либо создается заново, либо вводится из архива средствами языка Поляр и вырисовывается в рабочем окне экранного редактора в

установленном масштабе. Изображение отсекается по границам окна, это отсечение нельзя запретить. Взаимно-однозначное соответствие между структурой рисунка и его изображением автоматически поддерживается редактором в ходе сеанса работы. Размер рисунка на плоскости ограничен только максимальным абсолютным значением координат, зависящим от размерности машинного слова.

Масштаб, в котором рисунок изображается на экране, выбирается пользователем, структура рисунка при этом не изменяется.

Символы шрифта, используемого в системе, синтезированы из некоторого подмножества примитивов, возможно, объединенных в сегменты. Это позволяет работать с текстом так же, как и с произвольным рисунком: вращать, отражать, изменять его размер и цвет. Шрифт включает в себя стандартный набор символов. Кроме того, пользователь по своему усмотрению может определить собственные знаки с кодировкой, соответствующей управляющим символам, а также изменить изображение любого символа. При вводе текста символы набираются с обычной алфавитно-цифровой клавиатуры.

Набор операции в редакторе для манипуляций с рисунком: создание, копирование, сдвиг (параллельный перенос всех вершин), поворот на любой угол в градусах, отражение относительно оси, изменение размера, цвета, положения вершин прямоугольника, линии и многоугольника, уничтожение (создание) вершины на линии или многоугольнике.

Для указания места на рисунке реализованы два варианта курсора: небольшой мерцающий крестик или пересекающиеся вертикальная и горизонтальная линии, продолжающиеся до границ окна. А для слежения за положением курсора на рисунке используется понятие текущей точки. При передвижении по рисунку текущая точка автоматически поддерживается редактором в пределах экрана. Если она выходит за пределы, то рисунок на экране сдвигается.

Значения приращений координат текущей точки при нажатии управляющих клавиш (шаг перемещения по рисунку) устанавливаются в процессе работы.

Значения текущих координат курсора отображаются в специальном окне и могут быть изменены пользователем, после чего курсор устанавливается в указанную точку. Так осуществляется прямая адресация в абсолютных координатах.

Построение нового примитива начинается из текущей точки рисунка, т. е. из точки, в которой находился курсор в момент выбора команды построения, и завершается специальной командой.

Чтобы отредактировать примитив, необходимо выбрать его, сделать текущим. Для этого курсор подводится к вершине примитива и выполняется команда **выбор примитива**. Выбирается тот примитив, вершина которого оказалась ближе к курсору в момент выполнения команды выбора. Курсор становится на эту вершину. Если редактирующие действия не выполняются, а курсор сдвигается с вершины, то примитив теряется.

Примитивы строятся и редактируются так, что вершина, на которой стоит курсор, всегда остается вместе с ним. Остальные вершины либо остаются неподвижными, либо изменяют свое положение в соответствии с алгоритмом преобразования. Структура рисунка при этом модифицируется и все изменения сразу отображаются на экране.

Чтобы идентифицировать сегмент, с которым будет идти дальнейшая работа, необходимо выполнить команду **выбор сегмента**. При этом выбирается примитив, а сегмент определяется как минимальный, которому принадлежит выбранный примитив. Отметим, что выбранным считается также заново построенный примитив или сегмент.

Для удобства работы с редактором введены средства, характерные для текстовых экранных редакторов:

запись всего рисунка в архив до окончания сеанса редактирования;

механизм макрокоманд (т. е. имеется возможность определить некоторую непрерывную последовательность редактирующих действий как одну, новую команду). Макрокоманду можно записать в архив, прочитать из архива;

буфер редактора для временного хранения рисунков.

Запись в буфер происходит при выполнении трех различных команд: записи в буфер примитива, выделения фрагмента рисунка и чтения рисунка из архива. Координаты рисунка, находящегося в буфере, пересчитываются таким образом, что точка, в которой находился курсор в момент записи, переводится в начало координат.

При выделении фрагмента строится и модифицируется по общим правилам для построения примитивов прямоугольник. Все примитивы, попадающие в этот прямоугольник, образуют новый сегмент и запоминаются в буфере.

При чтении из архива выбранный рисунок записывается в буфер, он вырисовывается на экране и добавляется в структуру редактируемого рисунка. Координаты рисунка, выдаваемого из буфера, при этом пересчитываются так, что начало координат его совмещается с координатами курсора. С помощью этого механизма можно многократно копировать элементы рисунка.

Телефон 35-03-60, Новосибирск

ЛИТЕРАТУРА

1. Лельчук Т. И., Марчук А. Г. Язык программирования Поляр: описание, использование, реализация.—Новосибирск: ВУ СО АН СССР.—1986.—С. 94.
2. Алексеев Г. И., Лельчук Т. И., Марчук А. Г., Мыльников С. П. Переносимый транслятор с языка Поляр.—В сб.: Теория программирования и средства описания параллелизма дискретных систем.—Новосибирск: ВУ СО АН СССР.—1985.—С. 150—162.
3. Брябрин В. М. Интеллектуальный интерфейс на основе персональной ЭВМ // Микропроцессорные средства и системы.—1984.—№ 3.—С. 38—44.

Статья поступила 20.04.88

УДК 519.681.3.06

А. К. Дудолодов, Г. П. Епихов, Л. В. Ерохина,
А. Н. Малков, М. В. Минин, А. А. Федоров

ПАКЕТЫ ДЕЛОВОЙ ГРАФИКИ НА ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРАХ

По мере распространения персональных компьютеров (ПК), расширения сферы их применения, в частности использования для обработки числовой информации в сфере управления производством и сбытом продукции, выявления различных тенденций в коммерческой и других областях деятельности, все большее значение приобретает деловая графика. Подтверждение этому заключается не только в появлении на мировом рынке значительного числа пакетов деловой графики, но и в непрерывном возрастании ежегодного объема их продаж и в абсолютном исчислении, и в сравнении с другими видами программного обеспечения ПК. Обзор зарубежных пакетов деловой графики содержится в [2].

В СССР пакеты деловой графики в настоящее время в управлении и научных исследованиях не применяются, поскольку вычислительная техника с развитыми средствами графического представления данных практически отсутствует. Однако стремительно нарастающий выпуск ПЭВМ с цветными графическими мониторами и увеличивающиеся закупки техники из-за рубежа при-

водят к тому, что в нашей стране компьютеры уже оказываются на столах у управленцев, экономистов и т. д., и у них возникает потребность в деловой графике.

В настоящей статье описываются две программы деловой графики (ДГРАФ и График), реализованные в ВЦ АН СССР. Хотя приведенные программы и уступают западным прототипам по ряду параметров (табл. 1), но они обладают следующими очевидными преимуществами:

Таблица 1

Технические характеристики пакетов деловой графики

Наименование	Память, Кбайт	Максимальное число		
		столбцов	графиков	круговых диаграмм
Fasf Graphs	128	120	6	12
Microsoft Chart	128	512	512	64
Zebra	192	Определяет пользователь	12	12
Draw-It	192		100	100
CSS	256	Неограниченно	20	16
Grafit	128	15	5	Определяет пользователь
ДГРАФ	126	50	16	
График	27	10	10	—

полный контроль разработки вплоть до исходных текстов;

возможность интеграции с другими отечественными ППП, в которых требуется графическое представление числовых данных;

наличие русского алфавита (кириллицы);

возможность адаптации на отечественные ПЭВМ;

программа ДГРАФ является первой сданной в промышленную эксплуатацию отечественной программой деловой графики для ПЭВМ ЕС1841 и выше (под названием ДИАГРАММА) [3].

Программа графического представления данных ДГРАФ

Программа предназначена для промышленной эксплуатации на отечественных персональных ЭВМ типа ЕС1841 с объемом оперативной памяти не менее 512 Кбайт, цветным графическим дисплеем и матричным печатающим устройством под управлением операционной системы ДОС-16.

Программа ДГРАФ обеспечивает автоматическое масштабирование и построение на экране дисплея графических диаграмм трех типов: гистограмм (столбиковая диаграмма), линейных графиков и секторных диаграмм (рис. 1—3). Каждая из перечисленных диаграмм снабжается заголовком и наименованиями строк и столбцов. Для вывода алфавитно-цифровой информации на экран разработан шрифт с символами переменной ширины, что дало возможность, с одной стороны, печатать на одной строке экрана около 60 символов (вместо 40 при использовании стандартного шрифта) и, с другой стороны, в значительной степени сохранить наглядность стандартного шрифта.

Данные в программу ДГРАФ поступают из текстового файла, который создает либо пользователь с помощью любого текстового редактора, либо другая программа. Это позволяет использовать программу как автономно, так и в составе интегрированных систем.



Рис. 1. Гистограмма (несколько групп столбцов)



Рис. 2. Линейный график



Рис. 3. Секторная диаграмма

Информация в файле представляется в виде псевдодоков.

Таблица исходных данных не должна превышать 50 строк на 50 столбцов. В режимах линейных графи-

ков и столбчатых диаграмм на экран отображаются 16 строк на 50 столбцов, в режиме секторных диаграмм — три строки из первых 16 строк на 16 столбцов, представленные в виде трех кругов, разделенных на секторы.

Программа ДГРАФ позволяет создавать каталог графических объектов в отдельном файле. В нем вместе с исходными данными (таблицей численных величин, текста заголовка объекта и т. д.) запоминаются также цвета раскраски рисунка, текстовые и графические комментарии. При сохранении объекта в каталоге пользователь задает его имя (до 8 символов) и расширенный комментарий.

Программа ДГРАФ работает в двух режимах: сокращенный вариант — только просмотр графического объекта, полученного из текстового файла; полный вариант — просмотр графических объектов, полученных из текстового файла и сохраненных ранее в каталоге; сохранение файла в каталоге; сохранение в файле DOS в виде рисунка в формате пакета PC PAINT и вывод изображения на матричное печатающее устройство типа FX-80 и TOSHIBA.

Программа обладает дружественным интерфейсом, для управления требуются минимальные навыки. Пользователь работает в режиме выбора и указания, используя несколько функциональных клавиш.

Основное меню в полном варианте работы программы представляет собой комбинированное меню: на горизонтальной строке находятся исполняемые действия, а при переходе с помощью стрелок вправо-влево на конкретную позицию меню над последней возникает окошко со списком возможных значений аргументов для данного действия. Такое расположение управляющей информации очень компактно и удобно для пользователя в процессе обучения работы с программой и при ее эксплуатации. Активно применяется принцип умолчания для выбора наиболее часто используемых альтернатив.

Во всех основных состояниях программы (каталог, меню, просмотр диаграммы) предусмотрена возможность вызова подсказки с помощью одной и той же клавиши.

В настоящее время программа ДГРАФ функционирует в составе интегрированной системы Спектр, разработанной в ВЦ СССР [1]. В диалоге с табличным процессором пользователь выделяет интересующий его фрагмент в таблице числовых данных и затем, исполняя соответствующий пункт меню, получает на экране выделенные данные в графическом виде.

При реализации программы были использованы инструментальные пакеты, разработанные в ВЦ АН СССР: модельная база данных MEMORY (Е. Н. Веселов, С. Г. Сироткин) и Администратор Окон (Г. Г. Гнездилова, И. В. Ладошкин).

Авторы приносят благодарность аспиранту ВЦ АН СССР Р. Д. Шарма за оказанную помощь в разработке интерфейса программы с пользователем.

Программа графического представления данных График

Программа График позволяет отображать в виде гистограмм фрагменты данных электронной таблицы и базы данных интегрированного пакета Спектр [1]. Отличительная особенность программы — возможность работы на ПК с алфавитно-цифровым дисплеем, так как все графические построения обеспечиваются средствами псевдографики.

Фрагмент данных (строка, столбец или произвольная прямоугольная часть) электронной таблицы (табл. 2) выделяется цветом с помощью передвижения курсора, занимающего одну ячейку таблицы. Затем эта информация записывается в текстовый файл на виртуальном диске в оперативной памяти ПК, и вызывается программа График. Такая схема подключения, когда График является отдельной программой, нескомпилированной в единый код с системой Спектр, позволяет легко из-

Таблица 2

Электронная таблица

Продукты	1960	1970	1980	1985	1986
Мясо	4,400	7,200	10,200	12,400	12,800
Молоко	9,500	24,000	30,900	36,900	39,200
Картофель	4,000	6,000	8,000	9,400	10,100
Овощи	4,100	7,700	11,800	13,700	14,500
Фрукты	2,000	4,200	5,200	6,300	7,100

менять конфигурацию системы и использовать программу График автономно, указывая в качестве параметра имя текстового файла с данными.

При построении графиков каждому столбцу таблицы, так же как и в программе ДГРАФ, соответствует группа столбиков на диаграмме в количестве, определенном числом строк в выделенном фрагменте. Названия столбцов размещаются вдоль оси X и являются именами групп (рис. 4), а названия строк выводятся в отдельном окне в правой верхней части экрана и идентифицируют цвет, которым отображаются данные одной строки таблицы. Реализована возможность изменения порядка отображения данных на обратный — по столбцам. При этом график полностью перерисовывается и названия меняются местами.

С помощью процедуры автоматического масштабирования рассчитывается соотношение отображения чисел по оси Y исходя из разности максимального и минимального значений фрагмента данных, а ось X соответствует значению, близкому к минимальному. Таким образом, диаграмма отображает относительные значения числовых величин. Абсолютное значение, соответствующее одному столбцу диаграммы, выводится в специальном окне. Текущий столбик указывается курсором, и с помощью его передвижения можно последовательно просматривать абсолютные значения выбранных величин.

В зависимости от количества отображаемых данных автоматически изменяется толщина столбиков гистограммы. Максимальное число данных не превышает 60, что связано с ограничениями на допустимое число символов в одной строке экрана. Программа не предостав-



Рис. 4. Гистограмма (одна группа столбцов).

ляет средств изменения цветов и размещения частей изображения на экране.

Разработана версия программы График, в которой предусматривается возможность интерактивной работы с данными, рассчитываемыми по формулам в зависимости от нескольких параметров. Варьируя значениями одного или нескольких параметров с помощью меню в нижней части экрана, можно анализировать динамическое изменение соотношений величин на графике. Это позволяет быстро и эффективно исследовать задачи прогнозирования, распространенные в деловой сфере применения интегрированных пакетов.

Телефон 135-13-40, Москва

ЛИТЕРАТУРА

1. Брябрин В. М. Программное обеспечение персональных ЭВМ.— М.: Нзука, 1988.
2. Business Graphics: Giving Power to Presentations//PC Magazine.— June 1985.— Vol. 4, N 2.— P. 110.
3. Программа ДИАГРАММА.— Таллин: Эст. ПО ГКВТИ, 1987.

Статья поступила 14.03.88

УДК 681.3.06

Н. А. Маркова

ГРАФИЧЕСКИЙ СТАНДАРТ И ЕГО ВНЕДРЕНИЕ НА ПЭВМ

Многообразие существующих графических средств (аппаратных и программных) ставит перед прикладным программистом проблему, что выбрать, и зачастую приводит в тупик: ничего не подходит. Тогда программист разрабатывает свое графическое обеспечение, увеличивая тем самым многообразие, но в редких случаях открывая дорогу для других программистов. Выход из этого заколдованного круга (или тупика) международное сообщество выработало в начале 80-х годов: необходимо выделить базовый, не зависящий от прикладной области и программно-аппаратной среды, уровень графических систем. Этот вывод был документально закреплен в международном стандарте Graphical Ker-

nal System (GKS). Соответствующий стандарт СЭВ Ядро графических систем (ЯГС) вступает в силу с 1988 г.

В чем преимущества использования стандарта, почему, несмотря на рекомендательный статус (на нем нет грифа «Карается по закону»), его соблюдение целесообразно для большинства приложений машинной графики?

Очевидное преимущество состоит в том, что прикладная графическая программа, использующая стандарт, мобильна, т. е. она может взаимодействовать с разнообразнейшим графическим оборудованием, исполняться в различных программно-аппаратных средах. Важно отметить, что передовое графическое оборудование, и в частности «законодатель-

мод» в области графической техники «Тектроникс», уже сегодня включает аппаратную реализацию GKS.

Не менее важно, однако, то, что внедрение стандарта способствует распространению передовой технологии программирования, эффект которой можно сравнить с эффектом, произведенным некогда структурным программированием: сокращение сроков создания и модификации программ, улучшение их качества (рост надежности и удобства использования).

Рассмотрим основные положения графической технологии, основанной на стандарте ЯГС, а также проблемы внедрения реализации стандарта для ПЭВМ.

Графический стандарт определяет набор взаимосвязанных правил вывода, ввода и хранения графических данных. Его реализация — это программные или программно-аппаратные средства, функционирующие по этим правилам. Прикладная графиче-

ская программа обращается к средствам, реализующим стандарт, как к библиотеке подпрограмм. Оператор или «конечный пользователь» видит формируемое в результате их работы изображение и вводит графические данные с помощью устройств ввода (клавиатуры, светового пера, планшета и др.). То есть, определяемые стандартом правила имеют две стороны — для программиста и оператора.

Для всех распространенных языков программирования, в том числе Фортрана, Паскаля, Ады, Си, определены «привязки» — форматы обращения к функциям стандарта. Действия подпрограмм, реализующих стандарт, состоят в формировании изображения в соответствии с заданными параметрами и графическом вводе — преобразовании сигналов, полученных от устройств ввода в логические значения, передаваемые прикладной программе.

Графический ввод в ЯГС состоит из примитивов, в том числе обобщенного графического, под которым реализатор может понимать любой графический объект.

Вызывая примитивы, программист указывает только координаты точек, символическую строку для ТЕКСТА, массив цветов для МАТРИЦЫ ЯЧЕЕК. Конкретный вид изображаемых примитивов зависит от установленных к моменту их вызова атрибутов (цвета, типа, шрифта, ориентации и пр.) и параметров масштабных преобразований. Преобразования координат в ЯГС позволяют формировать изображение из независимо масштабируемых фрагментов (окон), осуществлять отсечение примитивов по границам окон, привязку всего изображения или его фрагментов к различным участкам поверхности экрана.

Примитивы можно объединять в именованные группы (сегменты). Сегмент как единое целое независимо от остального изображения может подвергаться аффинным преобразованиям (сдвигу, повороту, масштабированию, отражению), делаться видимым-невидимым, переименовываться, удаляться и подвергаться другим манипуляциям.

Графический ввод в ЯГС позволяет свести все многообразие физических устройств ввода к шести логическим классам: ВВОД ПОЗИЦИИ (пары координат), ВВОД ПОСЛЕДОВАТЕЛЬНОСТИ ПОЗИЦИИ, ВВОД ЧИСЛА, ВЫБОР АЛЬТЕРНАТИВЫ, УКАЗАНИЕ НА ОБЪЕКТ (возвращающее программе имя сегмента и идентификатор примитива, на который указывает оператор), ВВОД СТРОКИ. Взаимодействие с оператором при вводе включает:

вывод подсказки (приглашения);
эхо-отображение ввода, обеспечивающее визуальный контроль действий оператора;

фиксацию оператором определенных (конечных) значений, сопровождаемую оповещением — визуальным эффектом, позволяющим проследить правильность этого действия;

возможность как синхронной, так и асинхронной работы.

Для каждого класса устройств ввода предлагается несколько типов эхо-подсказки. Так, для ВВОДА ПОЗИЦИИ эхо — это следующее перекрестие, резиновая нить, резиновый прямоугольник, числовое представление координат. Для ВЫБОРА АЛЬТЕРНАТИВЫ (меню) эхо — текстовая или пиктограммная клавиша, функциональная клавиатура, текстовая директива, число (номер альтернативы) и др.

В различных реализациях ЯГС может быть обеспечена та или иная номенклатура эхо-отображений с большей или меньшей свободой выбора его атрибутов (например, цвета резиновой нити или размера текстовых символов).

Графические данные вместе с сопровождающей пользовательской (неграфической) информацией могут быть сохранены во внешней памяти — *метафайлах* для последующего, возможно, измененного воспроизведения в рамках данной реализации ЯГС либо передачи на другие установки, возможно, с другими реализациями.

В ЯГС применены интересные конструктивные решения, обеспечивающие простоту использования для неискушенных пользователей и богатство возможностей для квалифицированных программистов. Важнейшее из таких решений — многоуровневая система умолчаний. Например, все атрибуты имеют значения умолчания. Программист, желающий просто обеспечить различимость двух ломанных, устанавливает для них различные ИНДЕКСЫ СВЯЗКИ — номера в таблице, где набор связок тип — толщина — цвет задается по умолчанию. Наконец тот, кто хочет вводить примитив с конкретными точно определяемыми атрибутами, устанавливает связку.

Следует отметить, что освоение ЯГС в полном объеме, особенно аспекты, связанные с асинхронным вводом, представляет пока на практике существенные трудности. Прежде всего это связано с отсутствием популярного учебника и недостаточным преподаванием в вузах.

Эффективность внедрения стандарта зависит от рационального подбора дополнительных функций и наличия средств поддержки, облегчающих использование его реализации. Рассмотрим вариант решения этих вопросов в рамках реализации ЯГС для ПЭВМ — комплекса ГРИЛ*, реализующего функции ЯГС для ПЭВМ

типа IBM PC (ОС—MS—DOS) и «Электроника 85» (ОС—ИНМОС 85).

Пользователи ПЭВМ, эксплуатирующие готовые интегрированные пакеты, привыкли к такому удобному средству общения, как «редактирование формы» (заполнение/корректировка содержания бланка), позволяющему ввести сразу много параметров и обеспечивающему целостность их восприятия, возможность оперативного исправления ошибок и др. В ЯГС одновременно могут функционировать несколько устройств ввода. Однако реализация этой возможности на ПЭВМ затрудняется тем, что там, как правило, всего одно физическое устройство ввода — клавиатура. Непонятно, например, что означает нажатие клавиши «стрелка вправо» при двух работающих устройствах ввода — перемещение курсора при ВВОДЕ СТРОКИ или изменение координаты при ВВОДЕ ПОЗИЦИИ. В ГРИЛ одновременно функционирующие устройства можно объединить в одну форму. Ввод с клавиатуры связывается с текстовым полем формы — устройством ввода.

Заметим, что в отличие от готовых пакетов, где редактирование формы производится в алфавитно-цифровом режиме, в ГРИЛ форма не только может содержать «графический» ввод позиции, но и размещаться на экране, содержащем основное рабочее изображение. Это очень удобно для многих приложений, так как не вынуждает оператора переключать внимание. Конечно, на устройствах с низким разрешением, дабы не загромождать изображения, размер такой формы не должен превышать 7...10 устройств. Для заполнения сложных таблиц необходимо воспользоваться алфавитно-цифровыми средствами.

Для избалованного красивыми картинками пользователя ПЭВМ в ГРИЛ предлагается широкий набор видов эхо-отображений устройств ввода и регулирующих их возможных модификаций параметров. Используемые для эхо-отображения примитивы снабжаются полным набором атрибутов, например резиновая нить может быть красной штрихпунктирной, белой непрерывной и тому подобной, для литер редактируемого текста определены размеры, цвет и другие атрибуты.

Программист имеет возможность определять клавиши, используемые для фиксации ввода, и клавиши, используемые в качестве функциональной клавиатуры, т. е. не обязательно любой ввод кончать «возвратом каретки», а меню функциональной клавиатуры можно организовать на произвольных клавишах, а не только на стандартных Ф1...Ф10.

Предлагаемый ГРИЛ набор возможностей по организации ввода позволяет легко строить произвольные проблемно-ориентированные интер-

* В настоящее время распространяется разработчиками.

активные программы, конкурентоспособные по качеству диалога с готовыми закрытыми (и потому имеющими ограниченное применение) интегрированными пакетами.

Другие дополнительные возможности, представляемые ГРИЛ, связаны с обработкой растровой информации. ЯГС имеет две функции (МАТРИЦА ЯЧЕЕК и ЗАПРОСИТЬ МАСШИВ ПИКСЕЛЕЙ) для работы с растровой информацией. Однако для обработки больших растровых массивов (таких, как фотоснимки, карты, факсимильные документы) эти функции не эффективны. ГРИЛ включает в себя набор так называемых битблочных растровых операций, соответствующих проекту стандарта «Интерфейс машинной графики» (CGI), позволяющих быстро копировать прямоугольные фрагменты растра как визуализируемого, так и невизуализируемого, что обеспечивает эффективность полиэкранного режима работы.

В ГРИЛ реализованы различные режимы наложения при формировании изображения, которые применяются не только к растровым, но и к «обычным» векторным примитивам. Устанавливая, например, режим наложения ИСКЛЮЧАЮЩЕЕ ИЛИ, можно получать «полупрозрачные» картинки, нижние части которых могут быть восстановлены повторным выводом верхних частей.

Графические данные по ЯГС имеют два уровня хранения: примитив и метафайл. В ГРИЛ предусмотрена процедура, которая воспринимает специальную «пользовательскую» запись как указание на необходимость интерпретации метафайла с указанным именем. Таким образом, можно организовать иерархию метафайлов, что существенно экономит дисковое пространство при хранении изображений, состоящих из большого количества однотипных фрагментов.

Важная составляющая ГРИЛ — его технологическая оснастка, включающая средства обучения пользователя и автоматизированной подготовки данных для конкретных прикладных программ.

Средства обучения пользователя — это своего рода интерактивный графический редактор, обеспечивающий интерактивный интерфейс с функциями ЯГС. Пользователь вводит значения атрибутов и параметры преобразования и затем рисует примитивы. Получающиеся картинки могут лучше объяснить значение понятий ЯГС, чем тома текстовых описаний.

Средства подготовки данных существенно облегчают для пользователя процесс кодирования параметров функций ЯГС.

Наглядным и красочным является способ заполнения областей шаблонами. Ручное кодирование таких шаблонов — трудоемкая работа. В ГРИЛ

предложено инструментальное средство — растровый редактор шаблонов, позволяющий вводить новые шаблоны или корректировать старые интерактивно.

Свобода, предоставляемая ГРИЛ для параметризации устройств ввода, оборачивается сложностью кодирования необходимых параметров. Специальное интерактивное средство дает возможность подбирать их в диалоге. Это средство следует рассматривать как инструмент прототипизации. С его помощью за сеанс работы можно сконструировать диалог («болтушку»), продемонстрировать его закачку, который может не только высказать пожелания, определяемые личным вкусом (например, касающиеся расположения и цвета окон), но и лучше осознать предлагаемый ему программный проект в его основных концептуальных аспектах. Это позволит выявить противоречия и внести необходимые изменения в программный проект на ранних этапах его проектирования.

Наконец, ряд средств предназначен для настройки параметров ЯГС, зависящих от реализации: шрифтов текстов, представлений типов линий и маркеров. Все эти средства работают по принципу растровых редакторов.

Телефон 930-05-16, Москва

Статья поступила 03.03.88

(Ответ на письмо на с. 57)

УВАЖАЕМЫЙ СЕРГЕЙ ГРИГОРЬЕВИЧ!

Ваши замечания по поводу нашей статьи в № 2 за 1988 г. справедливы. Действительно, в ней не сказано, чем вызвана необходимость разработки ДФС.

Загнанная проблема представления данных (можно, конечно, сказать и знаний) на всех ее уровнях, начиная с «информационных ресурсов» тов. Громова*, проектирования баз знаний, объектно-ориентированных систем и кончая проблемой размещения нескольких файлов на одном носителе, чрезвычайно злободневна, обширна и ждет своего решения.

В том или ином виде он стоит перед любым пользователем: системный ли это администратор крупного ВЦ, надменный ли обладатель IBM PC или владелец БК-0010. Видимо, нет смысла останавливаться на наших соображениях на этот счет, скажу лишь только, что дискуссия в журнале по этому поводу (не смею надеяться на постоянный акцент) нашла бы много заинтересованных читателей. Она не только помогла бы «поднять среднего пользователя над его компьютером», но и, возможно, приблизила бы нас к построению единого подхода в представлении данных, помогла бы понять, каков должен быть «образ жизни информационных ресурсов».

Итак, причины разработки ДФС и предоставляемые ею по сравнению со стандартными средствами преимущества.

По поводу необходимости в изменениях размера узла. Основной целью разработки ДФС являлось стремление избежать как раз тех операций, с негативными последствиями которых Вы и боретесь: сжатий физи-

ческих носителей; трудоемкой и опасной работы по перераспределению памяти между логическими дисками.

Приведу такой пример. Предположим, что двумя программистами используется диск объемом в 40000 блоков, причем каждый из них имеет логический диск в 20000 блоков. Чтобы забрать у одного программиста 1000 блоков и передать другому, нужно выполнить вручную несколько манипуляций, требующих большого внимания, чреватых потерей данных при малейшем сбое ЭВМ или ошибке оператора. С использованием средств ДФС для этого достаточно 1—2 команд монитора, не требующих особых расчетов и не приводящих к перемещению пользовательских данных на диске. При этом даже вследствие останова ЭВМ потеря информации не будет.

По поводу необходимости средств защиты доступа. Этими средствами в ДФС можно и не пользоваться, поскольку они не подменяют, а, я бы сказал, органически дополняют средства РАФОС II.

Практика использования больших файловых систем показывает, что: применяемые методы и средства структуризации информации играют огромную, если не первостепенную роль;

структуризация информации, соответствующая структуре коллектива пользователей, предпочтительна при небольших ее (информации) объемах;

структуризация информации, соответствующая ее семантике, тем более предпочтительна, чем больше объем информации в файловой системе.

В реальных приложениях оптимальным, видимо, является смешанный вариант структуризации. Для реализации таких вариантов и разрабатывалась ДФС, позволяющая:

(Окончание см. на с. 94)

* Громов Г. Р. Национальные информационные ресурсы. — М.: Наука, 1984. — 237 с.

УДК 681.395.2

В. П. Ковальчук, В. М. Семчишин

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ ДЛЯ СИСТЕМЫ НА БАЗЕ КОНТРОЛЛЕРА «ЭЛЕКТРОНИКА МС2702»

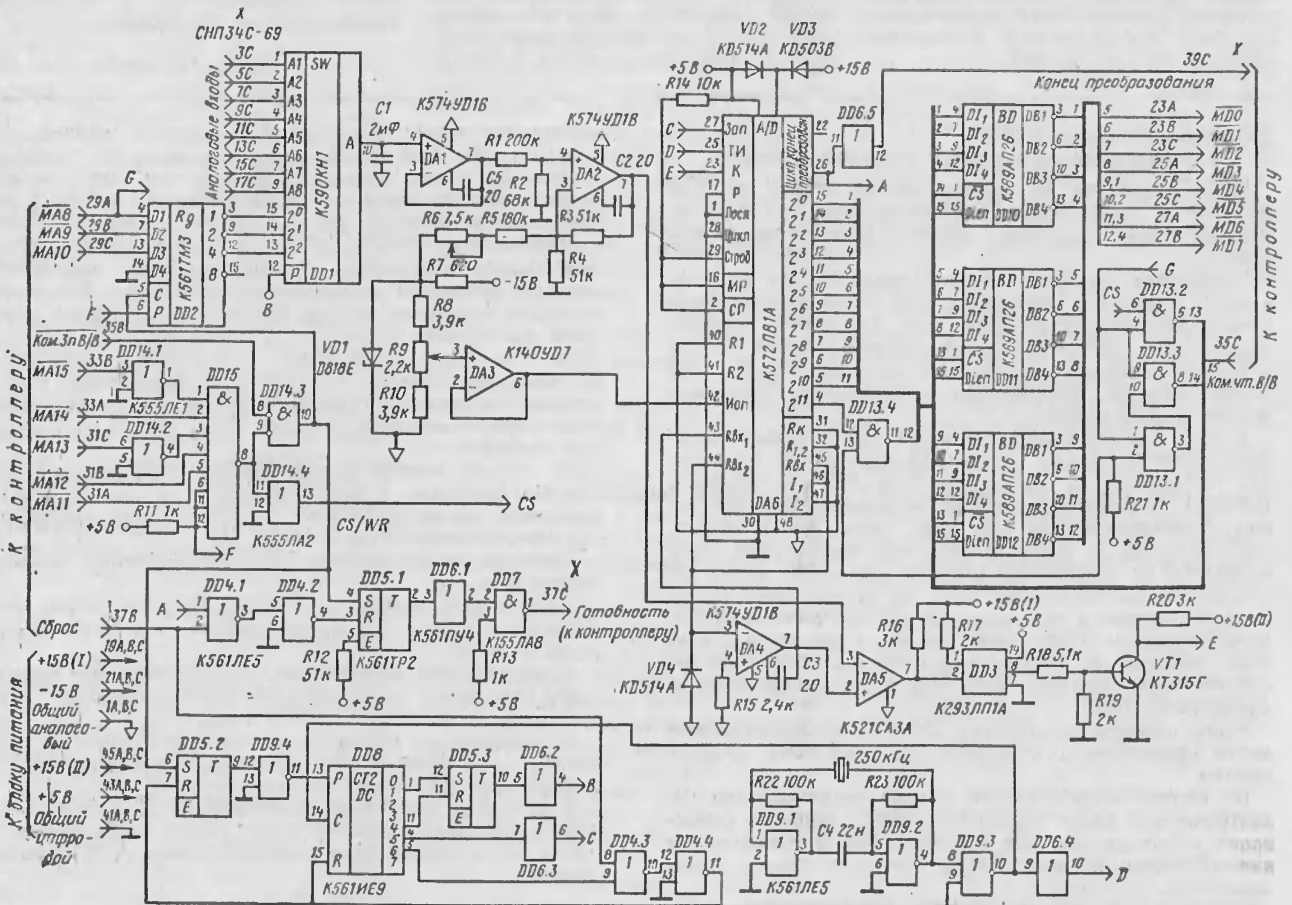
Контроллер «Электроника МС2702» не применяется для автоматизации объектов, характеризующихся не только дискретными, но и непрерывными входными и выходными параметрами из-за отсутствия в его комплекте аналого-цифровых и цифро-аналоговых преобразователей [1]. Разработанный авторами 12-разрядный аналого-цифровой преобразователь (АЦП) на базе БИС К572ПВ1, сопрягается с магистралью контроллера, предназначен для преобразования (см. рисунок) восьми аналоговых сигналов (напряжения переменного или постоянного тока) в цифровые коды и выдачи этих кодов на магистраль данных контроллера по командам микропроцессора (МП).

С помощью команды «Вывод» МП включает аналоговый канал (его адрес указывается во втором байте команды) и запускает АЦП. После запуска АЦП МП

контроллера может быть в двух режимах: в состоянии ожидания (если выход «Готовность» АЦП соединен с шиной «Готовность» контроллера) или в режиме выполнения программы (если такого соединения нет).

В первом режиме при выполнении команды «Вывод» МП выставляет на магистраль адреса адрес аналогового канала (A0H...A7H), напряжение из которого должно быть преобразовано в код. Пять старших разрядов адреса A11...A15 дешифруются дешифратором DD14.1, DD14.2, DD15, DD14.4, вырабатывающим сигнал выбора устройства CS. Три младших разряда адреса A8...A10 поступают на информационные входы регистра DD2. После этого МП выдает сигнал Ком.Зап.В/В, стробирующий сигнал CS, в результате чего в схеме вырабатывается сигнал CS/WR.

По фронту этого сигнала записываются три младших разряда адреса в регистр DD2. Выходной код DD2 подготавливает к включению соответствующий канал коммутатора DD1. Одновременно сигнал CS/WR, устанавливая триггер DD5.1 в Лог.1, переводит шину «Готовность» в Лог.0. При этом МП переходит в состояние ожидания. Сигнал CS/WR устанавливает в Лог.1 и триггер DD5.2, что приводит к появлению на входе P DD8 разрешающего потенциала, и счетчик DD8 начинает работать. При появлении Лог.1 на выходе 1



Принципиальная схема АЦП

DD8 триггер DD5.3 переходит в Лог.1, на входе Р аналогового коммутатора DD1 появляется Лог.1, что обеспечивает подключение закодированного в трех младших разрядах адреса А8... А10 аналогового канала ко входу СВХ С1, DA1, и происходит заряд С1 до уровня сигнала на входе.

Через три периода частоты генератора тактовых импульсов DD9.1... DD9.2 триггер DD5.3 переходит в Лог.0. В результате соответствующий ключ аналогового коммутатора размыкается, а на С1 образуется мгновенная выборка напряжения из соответствующего канала. С поступлением на вход С счетчика DD8 пятого импульса на выходе 5 DD8 появляется импульс длительностью 4 мкс, запускающий АЦП. АЦП работает по известному принципу поразрядного уравнивания с внешним компаратором DA5 и преобразователем токового сигнала ЦАП в напряжение DA4.

При поступлении на вход счетчика DD8 шестого импульса триггера DD5.2 устанавливается в Лог.0 сигналом с выхода 6 DD8 через логическую схему ИЛИ DD4.3 и инвертор DD4.4. Схема управления сбрасывается в Лог.0 и при подаче питания на контроллер, а также при нажатии на клавишу R пульта управления контроллером по шине «Сброс».

По окончании преобразования на выходе «Конец преобразования» DA6 появляется единичный импульс, сбрасывающий триггер DD5.1 в нуль, на шине «Готовность данных» — Лог.1. МП, выйдя из состояния ожидания, продолжает выполнять программу. Если далее в программе имеется команда ввода по адресу A0H или A1H, то данные из буферного регистра DA6 через шинные формирователи DD11, DD12 или DD10 передаются в магистраль данных контроллера.

Старший байт данных выбирается из АЦП с помощью команды «Ввод» по адресу A0H, а младший байт — «Ввод» по адресу A1H.

Выполняя эти команды, МП выдает на магистраль адреса адрес A0H или A1H. В результате дешифрации пяти старших разрядов вырабатывается сигнал CS. Этот сигнал и младший разряд адреса А8 поступают на логическую схему DD13.1, DD13.2, DD13.3, вырабатывающую сигналы выбора кристалла для шинных формирователей DD11, DD12 и DD10. Если А8=0, то активизируются шинные формирователи DD11, DD12, а если А8=1, то — DD10. При поступлении от контроллера сигнала «Ком. чт.В/В» соответствующий байт через шинные формирователи DD11, DD12 или DD10 передается на магистраль данных контроллера MD0... MD7.

Возможна работа АЦП и в режиме прерывания. Для этого шины «Готовность» АЦП и контроллера не соединяются между собой, а выход АЦП «Конец преобразования» подсоединяется к одному из входов «Запрос прерывания» контроллера.

Настраивается АЦП подстроечными резисторами R6 (настройка нуля) и R9 (настройка коэффициента преобразования). Для повышения помехоустойчивости аналоговая часть преобразователя гальванически изолирована от цифровой части с помощью оптрона DD3.

Погрешность АЦП определяется в основном температурным и временным дрейфами нуля операционных усилителей, дрейфом порога срабатывания компаратора, а также нестабильностью источника опорного напряжения [2]. Для уменьшения погрешности преобразования дрейф нуля АЦП корректируется программно: измерением нулевого напряжения в одном из аналоговых каналов и вычитанием из массива данных кода нулевого напряжения [3]. Цифровая коррекция дрейфа нуля позволила обеспечить требуемые метрологические характеристики преобразователя и уменьшить аппаратные затраты.

Диапазон преобразования АЦП — 5... +5 В. Двоичные числа на выходе АЦП представлены в дополнительном коде, а модуль числа — разрядами 1... 11 (12-й разряд — знаковый). Время преобразования — 120 мкс, относительная погрешность — 0,10 %.

Схему легко модифицировать для измерения однополярного напряжения в диапазоне 0... +5 В, исключив операционный усилитель DA2, инвертор DD13.4, а выход DA1 соединив со входом компаратора DA5.

282011, Тернополь, ул. Сливенская, 7, кв. 96, Ковальчуку В. П.

ЛИТЕРАТУРА

1. Контроллер программируемый универсальный «Электроника МС2702». — Техническое описание и инструкция по эксплуатации.
2. Шляндин В. М. Цифровые измерительные устройства. — М.: Высшая школа, 1981. — С. 194—220.
3. Блинов Н. А., Касьянов В. В., Паничев А. В. Помехоустойчивый АЦП, сопрягаемый с микропроцессором КР580ИК80//Микропроцессорные средства и системы. — 1985. — № 2. — С. 57—59.

Статья поступила 25.11.87

УДК 681.3—181.48 : 007.52

Н. Т. Данильченко, Л. Н. Лозовой

ПЕРЕНАСТРАИВАЕМАЯ МИКРОПРОЦЕССОРНАЯ СИСТЕМА УПРАВЛЕНИЯ САМОХОДНЫМ РЕНТГЕНОВСКИМ ДЕФЕКТΟΣКОПОМ «СИРЕНА-1»

Существующие методы контроля качества кольцевых сварных швов в ходе строительства магистральных трубопроводов высокого давления базируются на использовании портативных импульсных [1] и стационарных непрерывных [2] рентгеновских излучателей. Это полностью исключает возможность автоматизации процесса контроля в строящейся нитке трубопровода, так как при их использовании все операции по установке дефектоскопа, наложению рентгеновской пленки на контролируемый шов, включению излучателя выполняются

оператором. Это обусловлено прежде всего внешними условиями (характеристиками окружающей местности), часто не позволяющими наиболее эффективно установить дефектоскоп в зоне контролируемого шва, а также необходимостью производить от четырех до восьми снимков различных участков одного и того же сварного шва для получения полного панорамного изображения этого шва [3].

Преодолеть указанные препятствия можно, используя автоматизированные средства доставки рентгенов-

ского излучателя в зону контролируемого шва по внутренней поверхности трубопровода, что полностью исключает влияние внешних условий. Кроме того, при облучении кольцевого сварного шва изнутри трубопровода (с его продольной оси) можно получить полный панорамный снимок этого шва за одну экспозицию. Это определяет возможность автоматизации контроля и значительно повышает производительность аппарата.

Наиболее перспективный способ автоматизации неразрушающего контроля качества сварочных работ при современных широких масштабах строительства газо- и нефтепроводов — использование самоходных робототехнологических комплексов, перемещаемых (внутри трубопровода) автоматизированным электроприводом с автономным источником питания, управляемых через стенки труб с помощью специальных коллимированных

радиоизотопных командоаппаратов [4, 5]. Однако эти аппараты имеют «жесткую» программу функционирования, ориентированную на управление оператором при выполнении каждого снимка сварного шва, что препятствует созданию полностью автоматизированных рентгенографических комплексов.

Гибкая автоматизированная система контроля качества кольцевых сварных швов трубопроводов построена на базе самоходного импульсного рентгеновского аппарата «СИРЕНА-1» [5] и МПК БИС серии КР580 (см. цв. фото на вкладке в МП № 6, 1988 г.).

Аппаратные средства.

Система управления (рис. 1) дефектоскопом «СИРЕНА-1» содержит три локальные подсистемы, взаимодействующие друг с другом: сбора и предварительной обработки измерительной информации (1); логический микроконтроллер для реализации алгоритма управления дефектоскопом (2); периферийного оборудования для реализации управляющих воздействий от логического микроконтроллера (3).

Подсистема 1 включает в себя детекторы Д1 и Д2 ионизирующего излучения радиоизотопного командоаппарата; датчик напряжения (ДН) вторичных импульсных источников питания; датчик разряда секции автономного источника питания (аккумулятора) ДР1...ДР3.

Подсистема 2 (логический микроконтроллер) выполнена на основе микропроцессора КР580ИК80А

и включает в себя центральный процессор, интерфейсный блок связи с датчиками и исполнительными механизмами, реализованный на основе БИС КР580ВВ55, тактовый генератор КР580ГФ24 и блок перепрограммируемых запоминающих устройств с ультрафиолетовым стиранием К573РФ5 для хранения программы работы аппарата.

Подсистема 3 содержит исполнительные механизмы: автоматизированный электропривод с локальной системой управления перемещением аппарата «СИРЕНА-1» внутри нитки трубопровода; систему пуска и отключения импульсного рентгеновского излучателя типа МИРА-2Д [1]; таймер времени выдержки экспозиции; таймер дальности хода дефектоскопа при движении в глубину трубы.

Система управления робототехнологического комплекса «СИРЕНА-1» построена по магистрально-модульному принципу. Все элементы подсистемы 1 объединены единой системной шиной данных (СШД) и подключены к логическому микроконтроллеру через порт А адаптера КР580ВВ55. Порт В этого адаптера связан с системной шиной управления (СШУ), к которой подключены все исполнительные механизмы дефектоскопа. Внутри

логического микроконтроллера информация, получаемая с СШД и передаваемая в СШУ, поступает через шину данных микропроцессора. Обмен информацией между источниками и приемниками, а также логическим микроконтроллером проводится без адресации измерительных датчиков подсистемы 1 и приемников команд подсистемы 3. Это возможно благодаря небольшому количеству источников данных и приемников команд и унитарному кодированию информации в СШД и СШУ.

Информация о состоянии движения дефектоскопов и об обработке паузы безопасности перед включением рентгеновского излучателя поступает в СШД не от датчиков, а непосредственно в СШУ. Эти данные используются при выработке соответствующих команд в логическом микроконтроллере (рис. 2) на перемещение аппарата и включение излучателя, а также для управления периферийным оборудованием робототехнологического комплекса.

К полуавтоматическим режимам работы относятся: пуск дефектоскопа по направлениям ВПЕРЕД или НАЗАД; включение рентгеновского импульсного излучателя в момент обработки паузы безопасности.

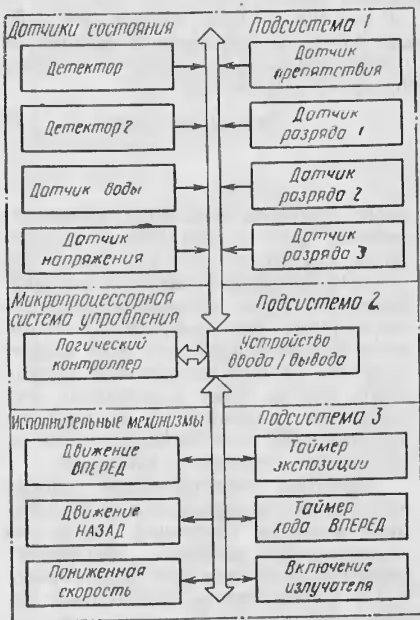


Рис. 1. Функциональная схема обработки данных в микропроцессорной системе управления аппаратом «СИРЕНА-1»

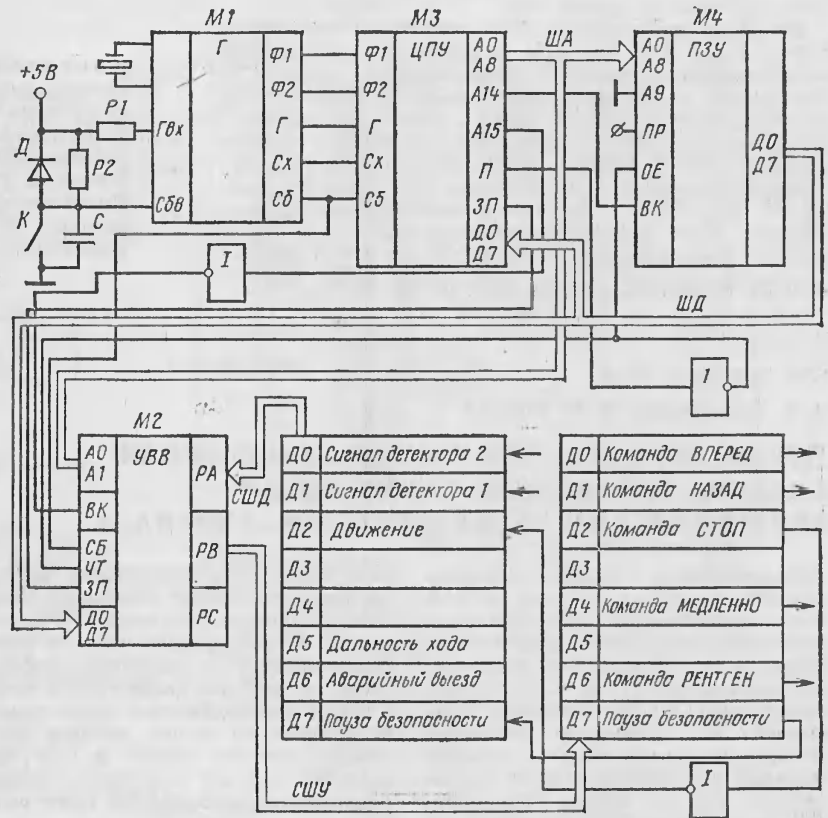


Рис. 2. Электрическая принципиальная схема логического микроконтроллера

В автоматическом режиме выполняются: перемещение дефектоскопа ВПЕРЕД и НАЗАД по внутренней поверхности нитки трубопровода на номинальной или пониженной скорости; остановка дефектоскопа в плоскости контролируемого сварного шва с высокой точностью; отсчет заданного времени экспозиции рентгеновского излучателя; блокировка системы управления и автоматический выезд дефектоскопа из трубопровода при снижении запаса энергии в источнике питания до заранее заданного уровня и при срабатывании датчика напряжения (при отказе вторичных импульсных стабилизаторов); остановка при отработке заданной установки времени таймером дальности хода. Усеченная система команд рентготехнологического комплекса «СИРЕНА-1» приведена в таблице.

Программные средства

Для реализации указанной системы команд целесообразен алгоритм, изображенный на рис. 3. Рассмотрим основные режимы работы.

Пуск аппарата. В исходном состоянии какие-либо команды отсутствуют и регистры общего назначения микропроцессора, используемые для хранения команд, находятся в нулевом состоянии. Команды подаются посредством перемещения радиоизотопного командоаппарата К с узкой диаграммой излучения над блоком детекторов (БД) слева направо при пуске ВПЕРЕД и справа налево при пуске НАЗАД (рис. 4). После команды на движение ВПЕРЕД или НАЗАД детекторы переходят в состояние $D1=0 \wedge D2=1$ или $D1=1 \wedge D2=0$, соответственно. Одновременно логическая функция Н принимает значение $N=1$ (соответствует первому циклу облучения детекторов) и микропроцессор фиксирует заданное направление движения аппарата. При дальнейшем перемещении командоаппарата К в том же направлении детекторы переходят в состояние $D1=D2=1$ (момент времени $T2$), а в момент времени $T3$ — в состояние $D1=1 \wedge D2=0$ при движении ВПЕРЕД или $D1=0 \wedge D2=1$ при движении НАЗАД. В обоих указанных состояниях робототехнологический комплекс «СИРЕНА-1» не изменяет своего положения, а полученные команды используются для более надеж-

Состояние детектора		Н	Реализуемая функция
D1	D2		
0	0	—	Ожидание очередной команды
0	0	1	Движение в заданную сторону с номинальной скоростью
1	0	1	Движение НАЗАД с номинальной скоростью
0	1	1	Движение ВПЕРЕД с номинальной скоростью
1	1	1	Движение в заданную сторону с пониженной скоростью
0	0	0	Включение рентгеновского излучателя
1	0	0	Движение ВПЕРЕД с пониженной скоростью
0	1	0	Движение НАЗАД с пониженной скоростью
0	1	0	Остановка дефектоскопа

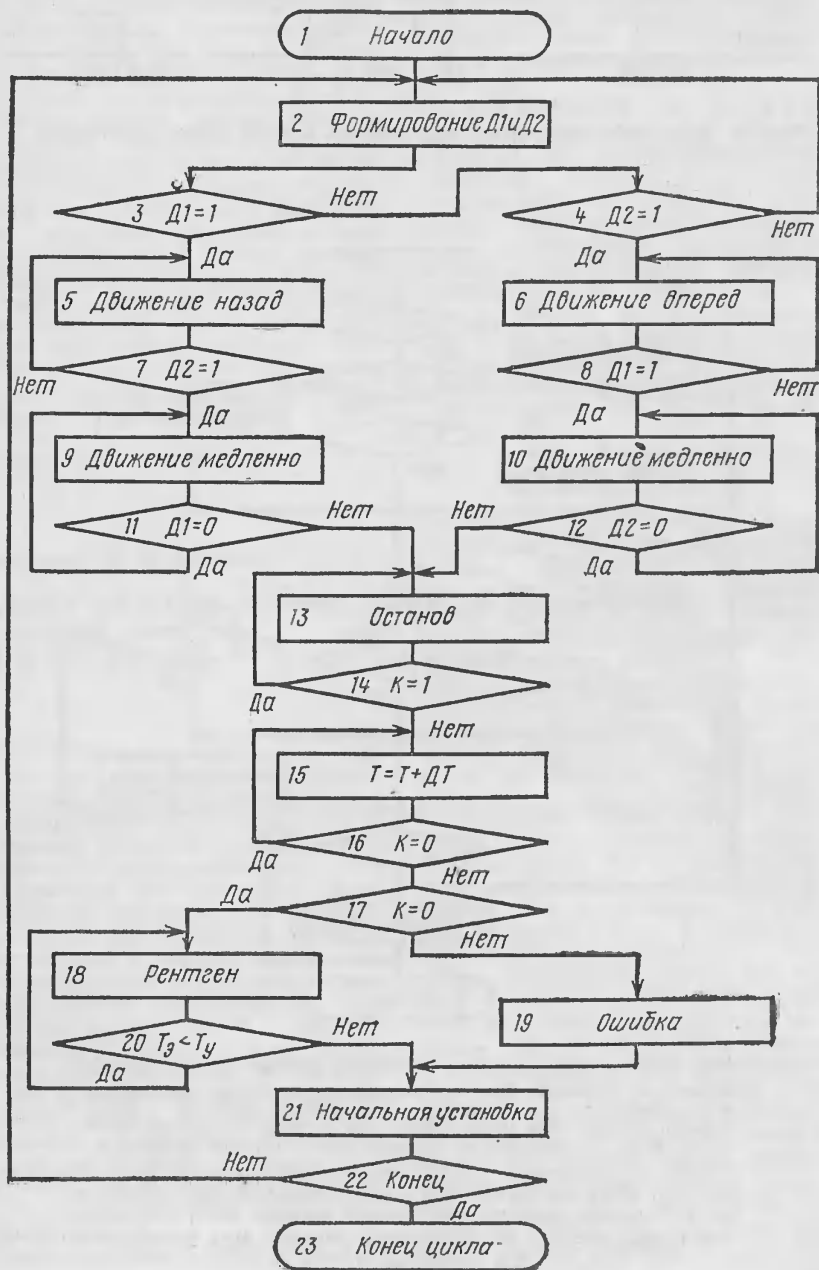


Рис. 3. Структурная схема алгоритма управления аппаратом «СИРЕНА-1»: Т — время выдержки паузы безопасности; DT — величина приращения времени Т; T_0 — текущее время экспозиции; T_y — установка таймера экспозиции. Изменение T_0 производится оператором 18, а начальное значение $T=0$ и $T_0=0$ задается оператором 13. $K=1$ при $D1=1 \wedge D2=1$, а $K=0$ при $D1=0 \wedge D2=0$.

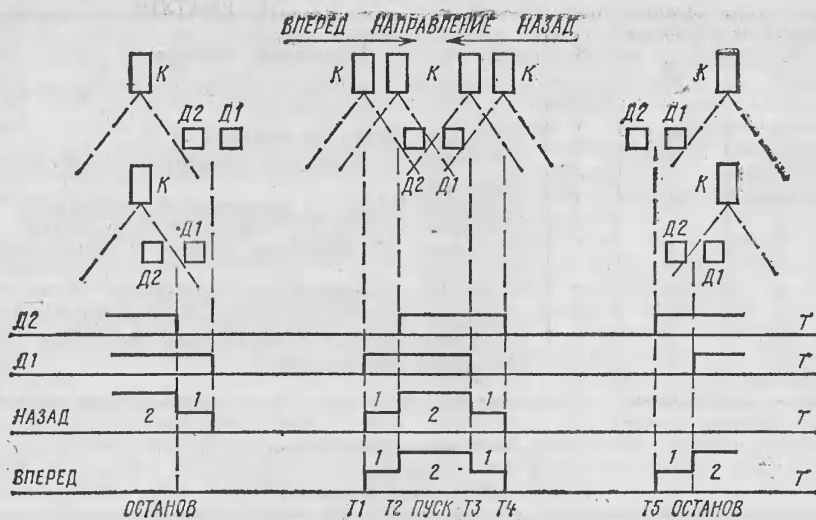


Рис. 4. Временные диаграммы кодирования команд блока детекторов

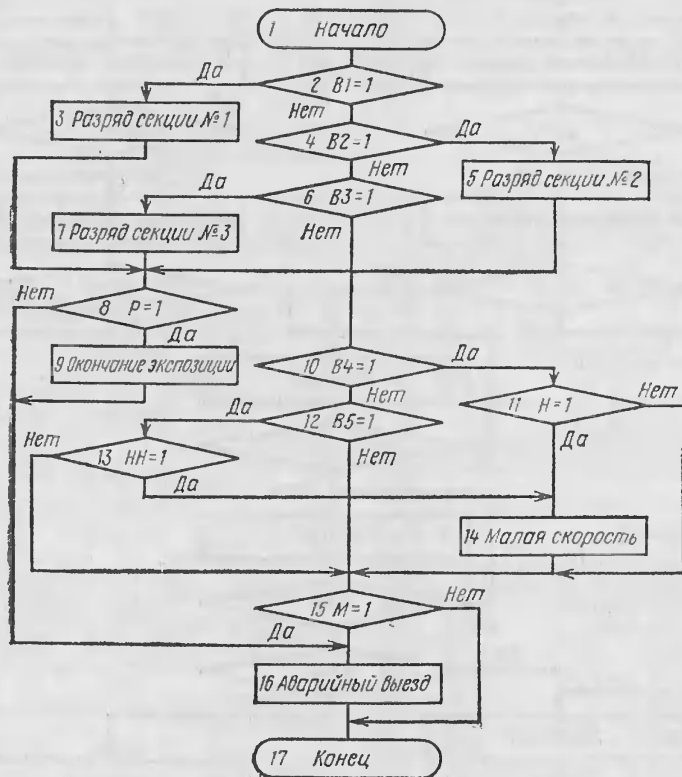


Рис. 5. Структурная схема алгоритма работы блока контроля. Логические функции: $P=1$ при включенном состоянии рентгеновского излучателя; $H=1$ при движении дефектоскопа ВПЕРЕД, а $HH=1$ — НАЗАД; $B1=1$ при $U_1 < U_y$; $B2=1$ при $U_2 < U_y$; $B3=1$ при $U_3 < U_y$; U_1 , U_2 и U_3 — напряжения на секциях аккумуляторной батареи, U_y — уставка по напряжению секций аккумуляторной батареи; $B4=1$ при $0 < \alpha < \alpha_y$; $B5=1$ при $\alpha_y < \alpha < 0$, α — угол наклона участка трубопровода, α_y — уставка срабатывания датчика наклона; $M=1$ при аварийном выезде аппарата из трубопровода, значение $M=1$ устанавливается в операторах 11, 13 и 14

ной идентификации команды на включение электропривода. В момент времени $T4$, когда оба детектора $D1$ и $D2$ не облучаются и имеют состояния $D1=D2=0$, электропривод включается на движение в заданную сторону и робототехнологический комплекс перемещается внутри трубопровода в заданном направлении с номинальной скоростью.

Остановка аппарата. Движение аппарата продолжается до внешнего управляющего воздействия командоаппарата K , установленного на поверхности трубопровода на определенном расстоянии от контролируемого шва. В этот момент при движении ВПЕРЕД состояние детекторов будет $D1=1 \wedge D2=0$, а при движении НАЗАД — $D1=0 \wedge D2=1$ (момент времени $T5$), т. е. порядок облучения детекторов изменяется на обратный по отношению к режиму пуска аппарата, а логическая функция H переходит в состояние $H=0$ (соответствует второму циклу облучения детекторов). Указанные состояния детекторов при $H=0$ приводят к торможению робототехнологического комплекса «СИРЕНА-1» и переходу электропривода на промежуточную скорость. Это обеспечивает повышенную точность остановки и устраняет упругие колебания в кинематических цепях электропривода. В момент возникновения состояния $D1=D2=1$ при $H=0$ самоходный робототехнологический комплекс останавливается в плоскости сварного шва.

Включение излучателя инициируется удалением командоаппарата K с поверхности трубопровода после остановки комплекса. Это переводит детекторы в состояние $D1=0 \wedge D2=0$ при $H=0$, микропроцессор отсчитывает паузы безопасности для удаления обслуживающего персонала на безопасное расстояние и автоматически включает рентгеновский излучатель и таймер выдержки экспозиции. В этом режиме контролируемый сварной шов просвечивается на рентгеновскую пленку. По окончании работы рентгеновского излучателя проводится начальная установка всех элементов системы управления в исходное состояние и они готовятся к восприятию новой последовательности команд.

Блок контроля внутреннего состояния и параметров окружающей среды (рис. 5) повышает надежность комплекса «СИРЕНА-1». Главная задача контроля внутреннего состояния — определение запаса энергии в автономном источнике питания аппарата. Если напряжение хотя бы одной секции аккумуляторной батареи ниже определенного значения, то самоходный комплекс в автоматическом режиме выезжает из трубопровода. Если это срабатывание произошло в момент рентгенографирования очередного

сварного шва, то аппарат выезжает из трубопровода сразу по окончании экспозиции.

Блок контроля измеряет и параметры внешней среды: углы наклона участков трубопровода, по которым аппарат перемещается; наличие непреодолимых препятствий механического типа (камни, «град» на сварных швах и т. д.); наличие влаги и водных препятствий. Причина возврата комплекса из трубопровода во всех случаях отображается на специальном светодиодном дисплее и регистрируется.

Структура программных средств аппарата «СИРЕНА-1» — модульная (МОНИТОР, ПУСК, ОСТАНОВ, ВКЛЮЧЕНИЕ ИЗЛУЧАТЕЛЯ, КОНТРОЛЬ и т. д.). Используемые при программировании принципы [6] позволили не применять аппарат подпрограмм и отказаться от ОЗУ для хранения стека. Применение в качестве ОЗУ регистров общего назначения микропроцессора значительно упростило аппаратную часть системы управления.

Программные средства обеспечивают: надежное декодирование и управляющих воздействий (исправление ошибок оператора при подаче команд, отмену ошибочно поданной команды включения рентгеновского излучателя и т. д.); приспособление к традиционным методам управления аппаратами подобного назначения и технологии рентгенографического контроля, что облегчает внедре-

ние и эксплуатацию аппарата в условиях строительства трубопроводов; надежное функционирование, не требующее от обслуживающего персонала специальных знаний и переподготовки дефектоскопистов.

Программные средства предоставляют дополнительные возможности: эффективную самодиагностику аппаратной и программной частей логического микроконтроллера; тестирование аппаратуры комплекса «СИРЕНА-1» по командам от специального выносного технологического пульта с клавиатурой и встроенным дисплеем, а также ряд других. В режиме тестирования проводятся следующие операции: ТЕСТ-ИЗЛУЧАТЕЛЬ — проверка работоспособности рентгеновского излучателя и таймера экспозиции (включение-выключение, правильность отсчета экспозиции и т. д.); ТЕСТ-ДАТЧИК — проверка исправности датчиков Д1 и Д2 детектора команд, датчика источника питания ДН, датчиков контроля напряжения аккумулятора ДР1...ДР3; ТЕСТ-ПРИВОД — проверка режимов ПУСК, ОСТАНОВ, выбор направления, включение, отключение и проверка правильности отсчета времени таймера дальности хода и т. д.

Комплекс «СИРЕНА-1» в настоящее время выпускается серийно. Одна из модификаций комплекса «СИРЕНА-1» экспортируется в Англию и другие развитые капиталистические страны.

Телефон 235-38-18, Ленинград

ЛИТЕРАТУРА

1. Месяц Г. А., Иванов С. А., Комяк Н. И., Пеликс Е. А. Мощные наносекундные импульсы рентгеновского излучения. — М.: Энергоатомиздат, 1983. — 168 с.
2. Рентгенотехника. Справочник под ред. В. В. Клюева. — М.: Машиностроение, 1981, с. 129—184.
3. Педанов Ф. Ф. Гамма-дефектоскоп ГДП-1 для панорамного просвечивания кольцевых сварных швов трубопроводов и других полых цилиндрических изделий // Дефектоскопия. — 1981. — № 11. — С. 38—42.
4. Данильченко Н. Т., Ершов Л. С., Лозовой Л. Н. и др. Самоходный импульсный рентгеновский аппарат СИРЕНА-1 // Дефектоскопия. — 1984. — № 7. — С. 98—100.
5. Данильченко Н. Т., Кристаллинский А. Л., Лозовой Л. Н. Автоматизированный комплекс СИРЕНА-1 для рентгенографического контроля качества сварных швов магистральных трубопроводов // Тез. докл. II Всесоюз. конф. молодых приборостроителей. — М., 1983. — С. 2.
6. Гладштейн М. А., Шубин Н. А. Об одном подходе к разработке программ для микропроцессорных вычислительных устройств // УСиМ. — 1984. — № 6. — С. 64—66.

Статья поступила 30.01.88

УДК 681.326

В. Ф. Матвеев, В. А. Никитюк, В. К. Прокопенко, П. И. Шевцов

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР НА БАЗЕ ОДНОКРИСТАЛЬНОЙ ЭВМ КМ1816ВЕ48 ДЛЯ ГРУППОВОГО УПРАВЛЕНИЯ СТАНКАМИ С ЧПУ В ГПС

При разработке ГПС механообработки необходимо организовать связь между системой управления ГПС и устройством ЧПУ для обмена управляющими программами, ведения технической диагностики оборудования, учета простоев оборудования по разным причинам и т. д. В соответствии с РТМ2 Н90—18—84 «Параметры сопряжения систем ПУ с ЭВМ и периферийными устройствами ввода-вывода» для сопряжения системы управления ГПС с устройствами ЧПУ выбран интерфейс радиальной последовательной связи (ИРПС).

Для ГПС в виде участков механообработки количество оборудования составляет 20...30 единиц, ГПС в виде цехов — до 100 единиц. Таким образом, комплекс технических средств автоматизированной системы управления ГПС включает в себя управляющий вычислительный комплекс

с 20...100 каналами ИРПС, к которым в качестве терминалов подключаются устройства ЧПУ. В соответствии с классификацией [1], такая сеть — локальная со своими уровнями функционального взаимодействия.

Разработка ПО протоколов взаимодействия локальной сети, включающей в себя до 100 каналов ИРПС, — это сложная и дорогостоящая задача. Альтернатива — использовать программируемый контроллер на базе однокристальной ЭВМ (ОЭВМ) КМ1816ВЕ48 [2], что позволит уменьшить количество каналов ИРПС в 6—7 раз и разместить ПО протоколов нижнего уровня (физического и канального) в ППЗУ ЭВМ КМ1816ВЕ48.

Программируемый контроллер (рис. 1) — это программируемый коммутатор и устройство связи (УС) с устройствами ЧПУ.

Программируемый коммутатор (ПК)

(рис. 2) состоит из однокристальной 8-разрядной ЭВМ (DD1—К1816ВЕ48), счетчика (DD2—К155ИЕ10), многорежимного буферного регистра (DD3—К589ИР12) и трех универсальных синхронно-асинхронных приемопередатчиков (УСАПП) (DD4...DD6—КР580ВВ51). ПК предназначен для приема информации кадра управляющей программы (УП) по входу 3 микросхемы DD4 (канал 1') из системы управления ГПС в последовательном коде, преобразования ее в параллельный код, записи в ОЗУ ОЭВМ DD1 и дальнейшей выдачи информации кадра УП в параллельном коде, преобразования ее в последовательный код с помощью УСАПП DD6 и выдачи в канал 3 (ИРПС) к УС с устройствами ЧПУ.

При тактировании работы ПК на вход 20 микросхем DD4...DD6 подаются высокочастотные тактовые импульсы частотой 1,9968 МГц, на вход 21 — сигналы установки в начальное состояние (RESET) и на входы 9 и 25 — тактовые импульсы с микросхемы DD2 для синхронизации данных (входы ТхС и РхС). Сигнал RESET обеспечивается программно и поступает на DD4...DD6 с разрядов 25...27 порта P2 ОЭВМ DD1. Скорость передачи данных 9600 бод.

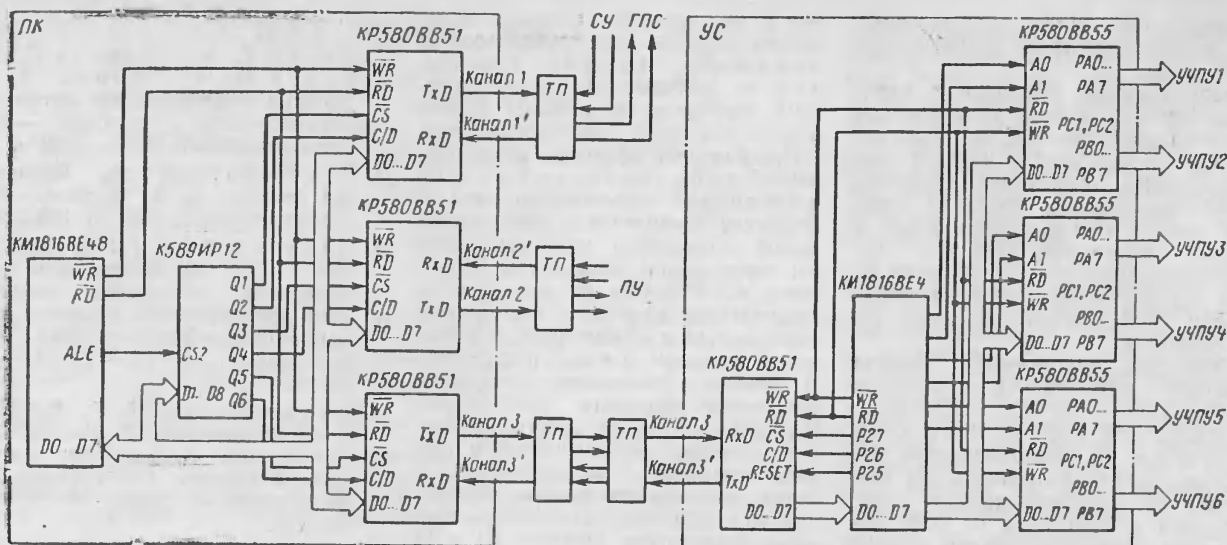


Рис. 1. Структурная схема программируемого группового контроллера

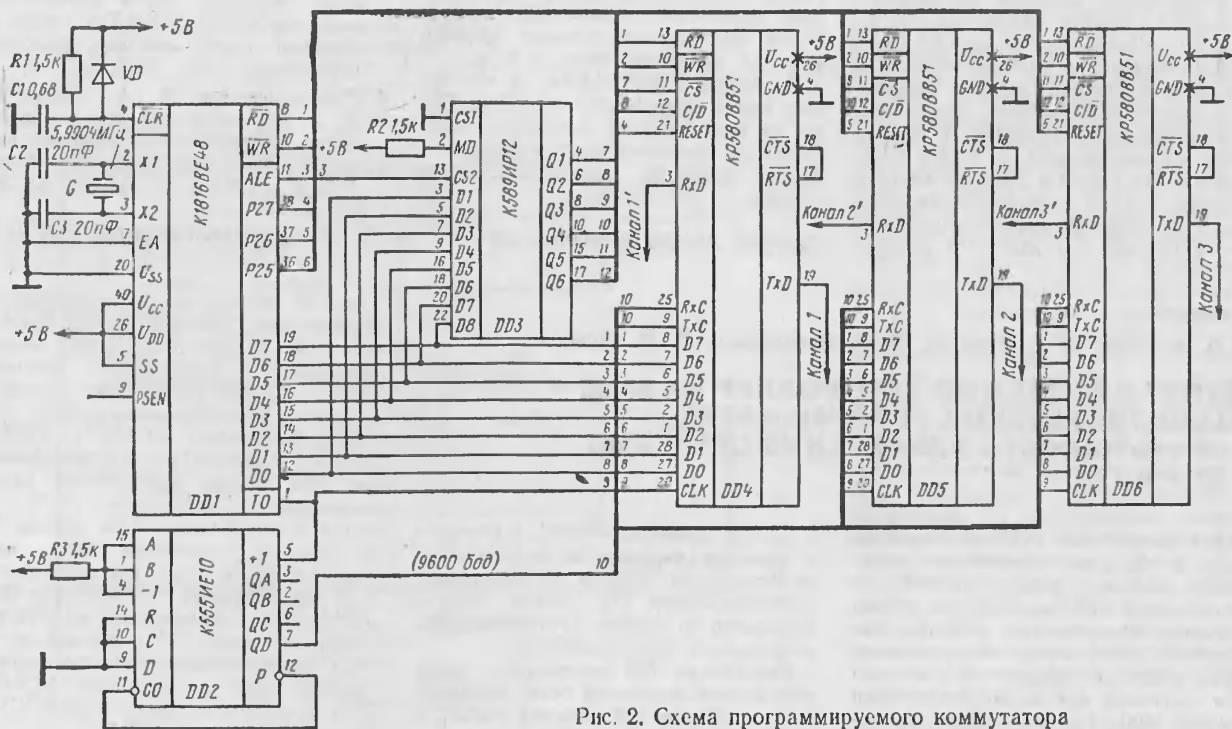


Рис. 2. Схема программируемого коммутатора

Каналы 2,2' (рис. 1) служат для подключения пульта (ПУ) группового управления. ПУ предназначен для выдачи в программируемый групповой контроллер запросов на трансляцию управляющих программ от системы управления (СУ) ГПС к устройствам ЧПУ.

Устройство связи (УС) с ЧПУ (рис. 3, 4) состоит из ОЗУ (DD1 — КМ1816BE48), счетчика (DD2 —

К155ИЕ10), УСАПП (DD4 — КР580BB51) и трех микросхем программируемого параллельного интерфейса (DD5...DD7 — КР580BB55). УС предназначено для приема информации кадра УП по каналу 3 (ИРПС) микросхемой DD3 — КР580BB51 (рис. 3), преобразование ее в параллельный код и записи в ОЗУ ЭВМ (DD1 — КМ1816BE48); считывания из ОЗУ ЭВМ информации кадра УП и

передачи ее на соответствующий выходной порт (рис. 4) микросхем (DD5...DD7 — КР580BB55). Сигналы портов микросхем должны соответствовать по уровню и физическому смыслу сигналам фотосчитывающего устройства (ФСУ), применяемого на конкретном устройстве ЧПУ, так как информацию кадра УП предполагается передавать по каналу ФСУ, программно обеспеченно-

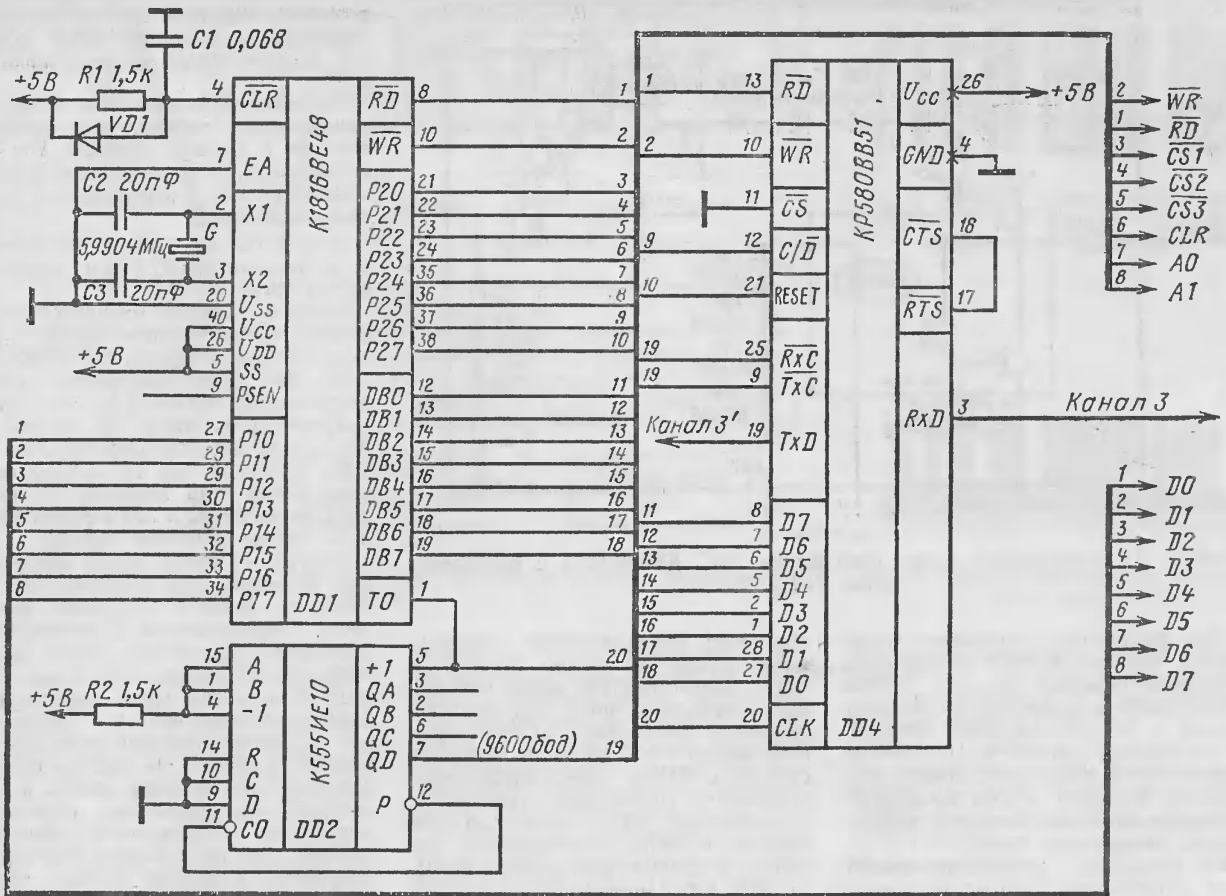


Рис. 3. Схема устройства связи с устройствами ЧПУ (часть первая)

му со стороны устройств ЧПУ.

Сигналы 1СП0...1СП7, 6СП0...6СП7 — информация кадра УП, соответствующая информационным сигналам дорожек перфоленты, соответственно для УЧПУ1...УЧПУ6. Сигналы 1СИ...6СИ — синхронимпульсы информации кадра УП, соответствующие импульсам синхродорожки перфоленты. Сигналы ГОТ СП1...ГОТ СП6 — сигналы готовности УЧПУ к приему информации, а ПУСК1...ПУСК6 — запроса на передачу информации кадра УП в УЧПУ. Сигналы СВОЙ1...СВОЙ6 — сообщение из ЧПУ в СУ ГПС о сбойной ситуации при передаче информации кадра УП (контроль по четности, по структуре кадра, запретным комбинациям).

Управление системой ЧПУ через выходной разъем фотосчитывающего устройства значительно проще, чем применение межинтерфейсных согласователей, и делает процесс управления независимым от типа применяемой системы ЧПУ. Таким образом, рассматриваемый программируемый контроллер рассчитан на передачу УП для устройств ЧПУ, имеющих в своем составе фотосчитывающее устройство для ввода управляю-

щих программ (например, 2Р22, 2У22, 2Р32, 2С42, 2С85, «Электроника НЦ 80-31» и др.).

Использование оптронов А1, А2, позволяет развязать входы и выходы устройств управления в «Токовой петле» приемного устройства. Це-

почка С1, R1, R2 применена для уменьшения фронтов импульсов. Диоды VD1...VD8 служат для термостабилизации режима работы оптрона, резистор R7 — для защиты от короткого замыкания в линии связи с ЧПУ.

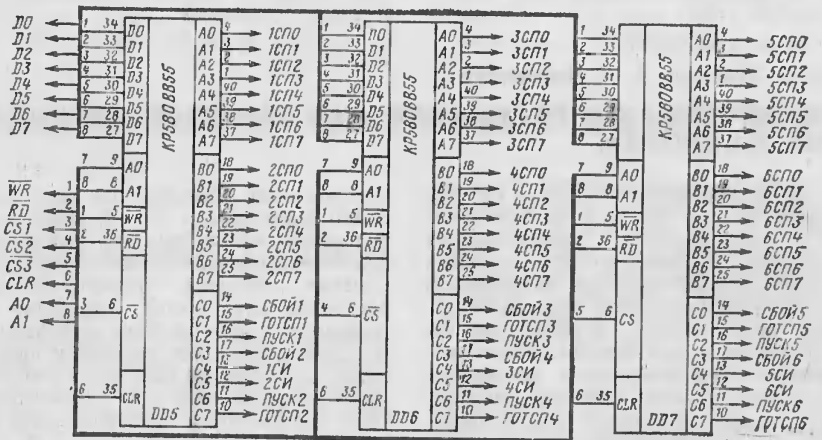


Рис. 4. Схема устройства связи с устройствами ЧПУ (часть вторая)

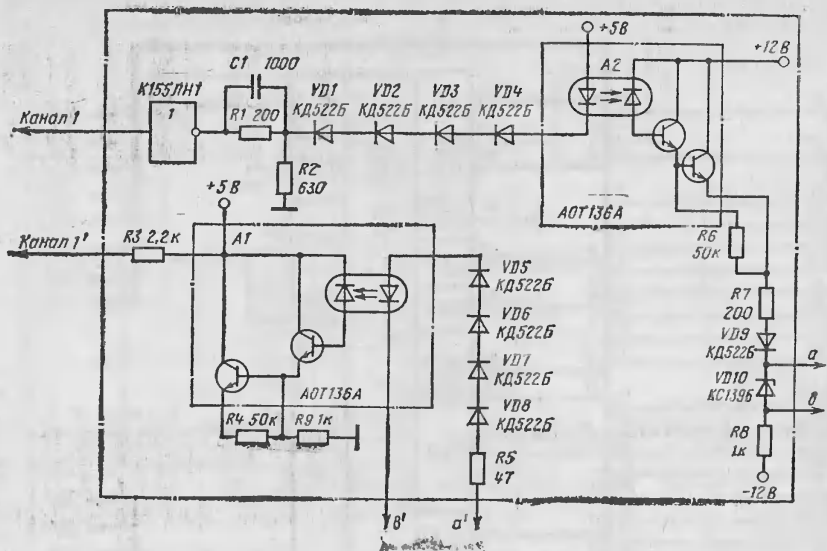


Рис. 5. Принципиальная схема сопряжения БИС КР580ВВ51 с каналами связи ИРПС

Для реализации управления физическим каналом ИРПС разработаны драйверы обмена согласно РТМ2 Н90—18—84 и включены на верхнем уровне в ОС РВ 3.0 ЭВМ СМ1420, а на нижнем уровне в ПО ЭВМ КМ1816ВЕ48. Управление физическим каналом включает в себя процедуры установления связи, передачи информации, завершения связи.

На канальном уровне локальной сети разработаны протоколы обмена согласно МУ 25741—86 «Автоматизированные системы управления ГПС. Каналы связи ЭВМ — УЧПУ. Технические средства, язык общения, программное обеспечение» для передачи управляющих программ. Согласно протоколам обмена программы обмена разработаны на верхнем и нижнем уровнях и включены в ПО ЭВМ СМ1420 и ЭВМ КМ1816ВЕ48.

Используя приведенную схему программируемого контроллера, можно

обслужить управляющими программами группу из шести станков с ЧПУ. МикроЭВМ ПК имеет возможность, используя порт P10...P17, наращивать количество УСАПП для сопряжения с УС до шести штук (36 станков с ЧПУ). При организации группового управления станками с устройствами ЧПУ цеха (до 100 станков с ЧПУ) используются три программируемых контроллера на базе ЭВМ КМ1816ВЕ48.

Телефон 36-54-27, Минск.

ЛИТЕРАТУРА

1. Якубайтис Э. А. Информационно-вычислительные сети. — М.: Финансы и статистика, 1984. — 232 с.
2. Кобылинский А. В., Липовецкий Г. П. Однокристалльные микроЭВМ серии К1816 // Микропроцессорные средства и системы. — 1986. — № 1. — С. 10—19.

Статья поступила 05.01.87

УДК 681.3.053+681.327

В. С. Дубровин, А. Д. Чернявский

МАРКИРОВКА И РАСПОЗНАВАНИЕ ШТРИХОВОГО КОДА НА МИКРОЭВМ

Вычислительная техника — удобное средство для учета и контроля самых разнообразных промышленных товаров. Однако в традиционных АСУП в ЭВМ вручную вводилась и затем обрабатывалась сопроводительная документация о промышленной продукции. Для максимального упрощения этого процесса во многих развитых странах широкое применение нашел штриховой код (определенная последовательность двухцветных полос различной толщины).

В статье рассмотрен аппаратно-

программный комплекс на базе серийно выпускаемых ПЭВМ ДВК, дополненный малогабаритной телекамерой «Электроника Н801» (рис. 1). Система позволяет генерировать на растровом печатающем устройстве с графическим интерфейсом штриховой код для маркировки товаров с качеством, достаточным для его устойчивого распознавания при считывании через телекамеру и преобразования в бинарное изображение.

Комплекс выполнен на базе ДВКЗ, канал которого расширен для под-

ключения вспомогательных модулей (использован дополнительный корпус [1], подключенный через коннекторы К5).

Разработано устройство ввода телевизионного сигнала для преобразования в форму, удобную для обработки на ДВК (последовательность машинных слов, передаваемых в канал микроЭВМ).

Устройство выполнено на полуплате размерами 135×240 мм, потребляет ток 0,5 А.

Связь с каналом микроЭВМ обеспечивается с помощью платы прямого доступа к памяти ИЗ (15КС-16-002). Для ДВК с временем цикла оперативной памяти 1,5 мкс было получено разрешение по строке на 288 точек, которые пересылались в ДВК блоками по 18 16-разрядных слов. Один кадр изображения передавался за 20 мс в 260 строках (брались только нечетные строки).

Программа ввода кадра изображения с телекамеры написана на ассемблере. Регистр состояния циклически опрашивается и проверяется готовность устройства. Затем в регистр адреса пересылается адрес массива М в оперативной памяти, а в регистр счетчика байтов — число байтов в дополнительном коде, участвующее в обмене. В регистр состояния заносится команда «старт» и циклически опрашивается готовность устройства к следующей операции обмена. До выставления готовности устройство в режиме прямого доступа к памяти пересылает N байтов в массив М. После этого управление передается вызывающей программе.

Для подготовки этикеток со штриховым кодом использовалось портативное печатающее устройство Роботрон СМ 6329.02.М, входящее в стандартную конфигурацию ДВКЗМ2. Программа, написанная на языке Паскаль, считывала файл с эталонными кодировками цифр от 0 до 9 в виде последовательности 0 и 1. В каждом коде 0 соответствовал полюсе единичной ширины белого цвета, а 1 — черного. В соответствии со стандартом [2, 3] использовалась кодировочная таблица для шестипозиционного кода. После этого оператор вводил шестизначное число, из которого программа формировала последовательность байтов непосредственно для управления иглами печатающей головки (рис. 2).

Процедура распознавания следующая. Этикетка со штриховым кодом считывалась через телекамеру в оперативную память (4 Кбайт). Далее проводился анализ 0 и 1 строки байтов, соответствующих строке считанного изображения. При этом значения каждого бита, определяющего белую или черную точку, распознавалось в массиве из 0 и 1. В этом массиве отмечались все переходы нулей и единиц, запоминались их координаты с одновременной фильтрацией

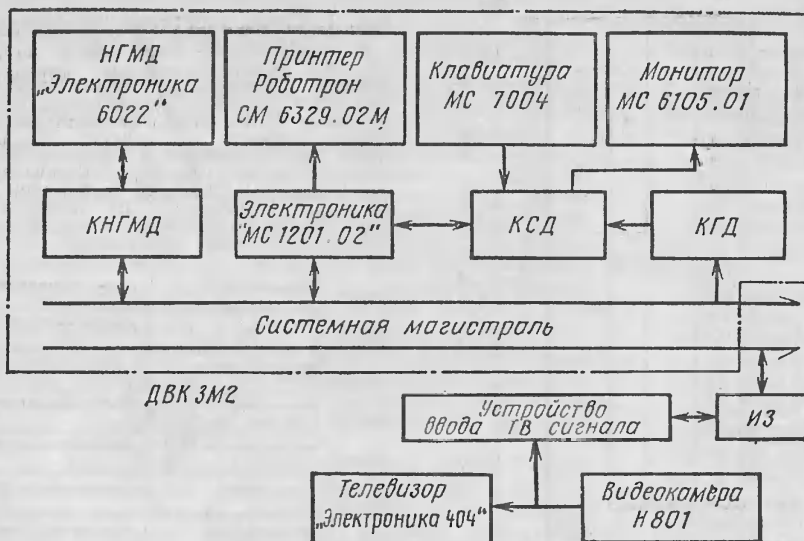


Рис. 1. Структурная схема комплекса для генерации и ввода штриховых кодов

одиночных помех. Ширина полос определялась по разности координат последовательных переходов (усреднялась по трем соседним строкам в середине изображения, а также по времени). Три раза проводился ввод одной и той же этикетки. В структуре штрихового кода известно относительное положение стартовых белых и черных полос единичной ширины, которые используются как эталонные и верхнее ограничение на ширину полос. Для определения ширины текущей полосы вычислялось отношение ширины полосы к эталонной γ и для $N=1..N_{max}$ проверялось неравенство $\gamma > N + \epsilon$. При его выполнении принималось решение, что ширина текущей полосы по крайней мере равна $N+1$. Алгоритм устойчиво работал при $0,2 < \epsilon < 0,6$.

В шестипозиционном штриховом коде каждая цифра кодируется четырьмя полосами. Вычисленная последовательность 0 и 1 сравнивалась поэлементно с эталонными последовательностями цифр от 0 до 9. При ми-

нимальном различии с одной из эталонных последовательностей, равно нулю, соответствующая цифра бралась в качестве очередной в распознаваемый штриховый код. При ненулевом отличии от всех эталонов фиксировалась невозможность распознавания и процедура повторялась.

Параллельно изображение со штриховым кодом наблюдалось на терминале ДВК (рис. 3).

Как показали эксперименты, при создании постоянного автономного освещения этикеток для распознавания каждой этикетки с вероятностью $>95\%$ достаточно усреднения не более чем по трем строкам и 3-кратного усреднения по времени. Вся процедура распознавания при этом не превышает 1,5 с.

Система разработана для использования в качестве входного устройства в управлении автоматизированной столовой. С ее помощью можно организовать питание в кредит, при котором каждый работник получает в пластиковом конверте уникальный

КОД145680



Рис. 2. Пример генерируемого штрихового кода для ширины единичной полосы в четыре точки.

КОД145684



Рис. 3. Пример введенного штрихового кода

штриховый код. При входе в обеденный зал код проверяется по таблице разрешенных и управляет открытием турникета.

Следует отметить, что разработанное устройство может быть использовано для ввода и анализа на ДВК произвольных черно-белых бинарных изображений.

Телефон 532-86-90

ЛИТЕРАТУРА

1. Малашевич Б. М., Романов Ф. И. Конструкция и компоновка диалоговых вычислительных комплексов // Микропроцессорные средства и системы.— 1986.— № 4.— С. 25.
2. Стандарт ДИН 66236. Структура штриховых кодов для машинной идентификации.
3. Ulf Lindqvist, Hanny Linna, Pertti Moilanen. The use of bar codes for materials handling in a printing plant 19-th IARIGAI Conf. Eisenstadt.—Austria, 1987.

УСТРОЙСТВА СВЯЗИ МИКРОЭВМ С ОБЪЕКТАМИ

УДК 681.323

Н. Н. Гаврикова, В. П. Островский

УНИВЕРСАЛЬНЫЙ АДАПТЕР КАНАЛА ВВОДА-ВЫВОДА ПЭВМ «ИСКРА» 226

Предлагаемый адаптер позволяет значительно расширить область применения ПЭВМ «Искра 226». С его помощью к каналу ввода-вывода ПЭВМ «Искра 226» подключаются устройства с нестандартными интерфейсами, для которых необходимы

управляющие воздействия (программаторы, схемные эмуляторы, устройства контроля программ, блоки памяти, технологические пульты контроля и т. п.).

Конструктивно адаптер выполнен на стандартной плате блока интер-

фейсного функционального (БИФ) и подключается к одному из рабочих мест «корзины» ПЭВМ. Канал ввода-вывода ПЭВМ «Искра 226» трансформируется в информационно-управляющую магистраль, содержащую информационные сигналы (D0...D7) и управления: \overline{RD} , \overline{WR} — сигналы чтения-записи источников-приемников информации; $\overline{H\bar{U}}$, \overline{AO} — стробирование выбора; \overline{RESET} — аппаратного сброса, формируемого при включении напряжения питания; $\overline{A0}$, $\overline{A1}$ — пря-

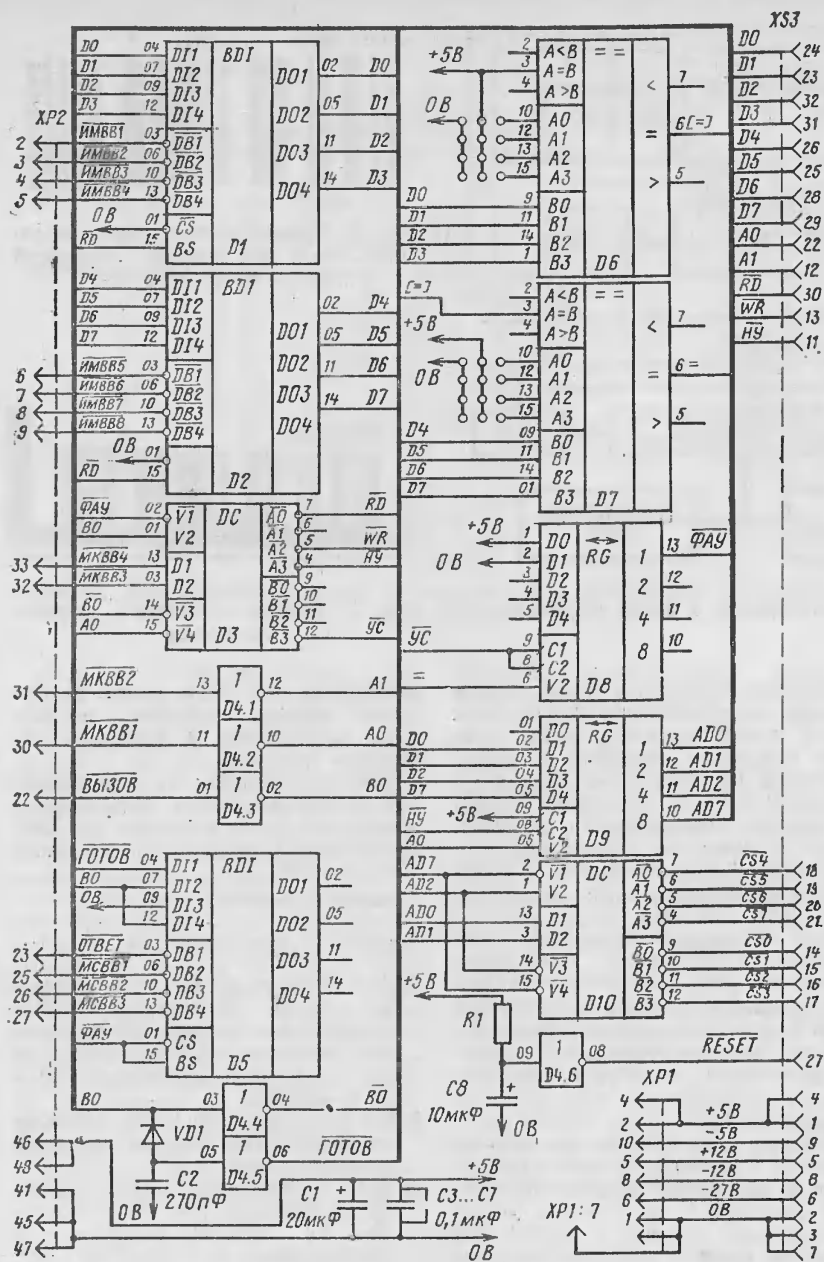


Рис. 1. Схема электрическая принципиальная адаптера:

D1, D2 (K589AP16) — буфер информационной магистрали; D3 (K155ИД4), D4.1, D4.2 (K155ЛН1) — дешифратор микрокоманд; D4.3...D4.5, D5 — узел синхронизации обмена по каналу; D6, D7 (K555СП1), D8 (K155ИР1) — селектор физического адреса устройства (физический адрес данного устройства набирается с помощью перемычек); D9 (K155ИР1), D10 (K155ИД4) — селектор источников информации

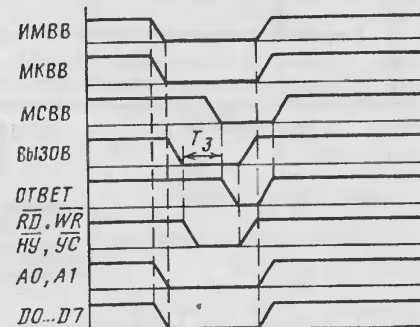


Рис. 2. Временная диаграмма работы канала ввода-вывода ПЭВМ «Искра 226»

мой адресации регистров; CS0...CS7 — выбора источников-приемников информации (пользователем могут не применяться).

На плате адаптера установлен разъем для подключения съемной платы размером 100×220 мм, определяющей его функциональное назначение (рис. 1). На разъем выводится информационно-управляющая магистраль, а также напряжения питания ПЭВМ «Искра 226»: +5 В (1,80 А), -5 В (0,25 А), +12 В (1,0 А), -12 В (0,5 А), -27 В (0,2 А).

Обмен данными по каналу ввода-вывода на языке БЕЙСИК осуществляется с помощью оператора \square GIO; используются системы команд канального процессора и программы CHANNEL и SERVICE. При разработке драйверов обмена следует обратить внимание на применение команды HY для адресации источников-приемников информации — перед ее выполнением в зону буфера канала должен быть занесен адрес HEX (00...FF).

Временная диаграмма работы канала представлена на рис. 2.

В настоящее время разработаны съемные платы для адаптера, на которых реализованы следующие устройства: программатор для программирования микросхем серии K573, схемный эмулятор МП K580ИК80, устройство контроля программ микропроцессорных систем. Возможны ограниченные договорные поставки адаптера и съемных плат, а также разработка заказных съемных плат. Срок разработки 2 месяца.

Телефон 01-02-77. Смоленск
Статья поступила 02.10.87

Соответствие команд канального процесса сигналам управления адаптера

Команды канального процессора	Сигналы управления				
	AO	A1	RD	WR	HY
Начальная установка (HY)	1	0	1	1	0
Принять первый байт (ППБ)	0	0	1	0	1
Принять байт (ПБ)	1	0	1	0	1
Принять байт последний (ПБП)	0	1	1	0	1
Принять команду (ПК)	1	1	1	0	1
Выдать первый байт (ВПБ)	0	0	0	1	1
Выдать байт (ВБ)	1	0	0	1	1
Выдать байт последний (ВБП)	0	1	0	1	1
Выдать уточненное состояние (ВУС)	1	1	0	1	1

УЗЛЫ ИНТЕРФЕЙСА ИРПС ДЛЯ ППЭВМ СЕРИИ «ИСКРА»

Интерфейс для радиального подключения устройств с последовательной передачей информации ИРПС [1] используется для дистанционной связи между устройствами ввода-вывода и контроллерами (адаптерами) ЭВМ. Передача в ИРПС осуществляется посылками постоянного тока 15...25 мА по двум двухпроводным линиям связи. В соответствии с [1] максимальная дальность передачи составляет 500 м при скорости передачи до 9600 бит/с, падение напряжения на входе приемника — не более 2,5 В. Устройства, соединяемые по ИРПС, гальванически развязаны. Развязка организуется со стороны устройства, не питающего ток линию связи, номинальное значение изоляционного напряжения — не менее 500 В [2—5].

Предлагаемые малогабаритные узлы ИРПС обеспечивают повышение дальности передачи до 2 км при скорости передачи до 9600 бит/с. Эти узлы входят в состав адаптеров аппаратуры передачи данных для ППЭВМ серии «Искра». Конструктивно ИРПС содержит узел источника токовых сигналов, включающий генератор тока, и узел приемника токовых сигналов. В качестве генератора тока могут использоваться генераторы нестабилизированного или стабилизированного тока.

Первый представляет собой либо усилитель тока [2], либо микросхему с открытым коллектором и ограничительным резистором [3]. Его применение несколько сужает диапазон допустимых изменений дальности передачи сигналов, что часто неудобно в эксплуатации.

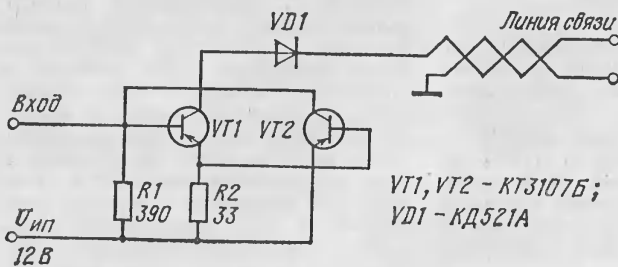


Рис. 1. Принципиальная схема генератора тока ИРПС

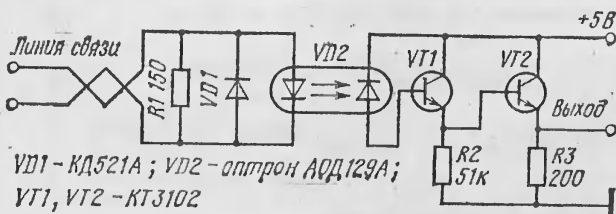


Рис. 2. Принципиальная схема приемника токовых сигналов

На рис. 1 предложена схема генератора стабилизированного тока со следящей обратной связью (модификация схемы «токового зеркала») [6]. В генераторе единственный источник питания 12 В и малое падение напряжения на транзисторе выходного каскада.

Приемники токовых сигналов могут быть разделены на две группы:

— приемники с оптроном АОТ 102Б, удовлетворяющим требованию гальванической развязки;

— приемники с оптронами других типов (АОТ 101Б, К293ЛП1А), не удовлетворяющие указанному требованию.

Оптон АОТ 102Б отличается малым уровнем выходного сигнала переменного тока (1,1—1,5 В) при большом разбросе положения рабочей точки на выходной характеристике, что затрудняет согласование выхода оптрона со следующим каскадом усилителя. Для согласования приходится вводить подстроечный элемент, смещающий положение рабочей точки на выходной характеристике оптрона, что увеличивает объем оборудования и приводит к необходимости подстройки при эксплуатации [5].

Использование для дистанционного обмена данными оптронов типа АОТ 101Б, К293ЛП1А, не удовлетворяющих требованиям к напряжению гальванической развязки, с нашей точки зрения, недопустимо по соображениям безопасности обслуживающего персонала.

В предлагаемой схеме приемника токовых сигналов (рис. 2) использован оптон АОД 129А с напряжением гальванической развязки 1000 В. Приемник содержит два каскада усиления, необходимые для повышения стабильности характеристик узла и сопряжения его с ТТЛ-схемами. Лабораторные испытания приемника показали высокую стабильность порога его срабатывания. Падение напряжения на входе оптрона АОД 129А не превышает 1,5 В. Совместное использование разработанных узлов источника и приемника токовых сигналов обеспечивает дальность передачи согласно протоколу ИРПС свыше 2 км.

Телефон 531-12-10, Ленинград

ЛИТЕРАТУРА

1. Средства вычислительной техники. Система малых ЭВМ. Интерфейсы. ЦНИИТЭИ приборостроения. Каталог. ГСП. ТС-2. — М., 1983. — Вып. 10, 11. — С. 81.
2. Корольков А. А., Раденко М. Е., Сенников В. К. Применение БИС КР580ВВ51 для реализации последовательных интерфейсов микропроцессорных систем // Микропроцессорные средства и системы. — 1985. — № 1. — С. 82—85.
3. Петух А. М., Романюк Р. Н., Подольский О. А. Преобразователь интерфейса ИРПС. — «Общая шина». // Микропроцессорные средства и системы. — 1985. — № 4. — С. 81—82.
4. Баронец В. Д. Микросерийные устройства дискретного ввода, вывода для микроЭВМ // Электронная промышленность. — 1985. — №9. — С. 3.
5. Мультиплексор передачи данных МПД СМ 8514. Инструкция по эксплуатации. 3.057.201 ИЭ.
6. Хоровец П., Хилл У. Искусство схемотехники, т. 1 / Пер. с англ. — М.: Мир, 1984.

Статья поступила 10.04.87

МНОГОФУНКЦИОНАЛЬНЫЙ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЙ КОМПЛЕКС НА БАЗЕ ОДНОПЛАТНЫХ ЭВМ

Функциональные возможности комплекса определяются набором плат, размещаемых в конструктиве

(БМО2) и объединяемых единой межмашинной магистралью (шина Multibus). Работа шины не зависит

от типа и разрядности процессоров на платах и от числа процессорных плат. Это позволяет организовать многопроцессорную работу и при распараллеливании вычислительного процесса превзойти производительность крупных ЭВМ. Кроме того, пользователь имеет полную свободу компоновки плат в БМО2 при первоначальном использовании МСУВТ

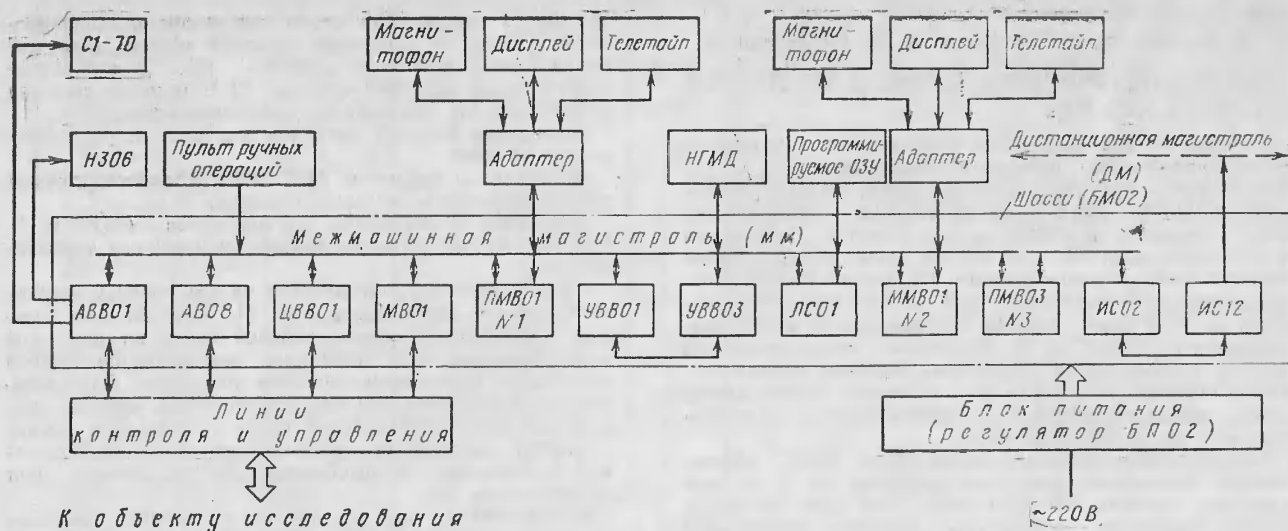


Рис. 1. Структурная схема управления процессом исследования ИУК МСУВТ В7

[1, 2] и при постепенном наращивании мощности и изменении конфигурации комплексов.

В данном комплексе применяются три процессорные платы (см. рис. 1) на базе МП КР580ИК80А, оснащенного наиболее полным и перспективным ПО, средствами отладки и документацией. Кроме процессорного элемента на платах расположены БИС параллельного (КР580ВВ55) и последовательного (КР580ВВ51) интерфейсов, контроллеры прерываний (КР580ВН59), обрабатывающие до 15 уровней прерываний, по два программируемых интервальных таймера (КР580ВИ53) (шесть реверсивных двоичных счетчиков), работающих в любом из шести режимов, блок управления межмашинной магистралью, а также ОЗУ и ПЗУ с общей емкостью 32 Кбайт. Большая часть БИС размещена на специальных разъемах. Это облегчает их замену при ремонте и настройке микросистем. Конфигурация таймеров, контроллеров прерываний и остальных БИС выбирается произвольно, по желанию пользователя, и реализуется с помощью проводных соединений, выполненных методом микропайки на платах.

Плата ПМВ01 (процессор № 1) — процессор интерфейса с объектом исследования, ПМВ01 (процессор № 2) используется для организации АРМ-П (автоматизированного рабочего места программиста), ПМВ03 (процессора № 3) — для связи с ЭВМ более высокого иерархического уровня с помощью ДМ (дистанционной магистрали).

Кроме того, в состав информационно-управляющего комплекса (ИУК) входят платы: арифметического расширителя (ПСВ01), расширителя па-

мяти (МВ01), ввода-вывода аналоговых сигналов (АВВ01 и АВ08), программатор кристаллов ПЗУ различных типов (ЛС01), контроллеры накопителя на ГМД двойной плотности (УВВ01, УВВ03), платы связи с ДМ (ИС02, ИС12), блоки электропитания комплекса (БП02).

Состав периферийного оборудования (рис. 2): дисплей (РИН609), два телетайпа (РТА80), НГМД (ЕС5079), два НМЛ (битовой магнитофон «Электроника-302»), графопостроитель (Н306), осциллограф светолучевой (С1—70), индикатор (ПИУ-1), пульт ручных операций, на который выведены все 15 уровней прерываний и кнопка возврата программ в исходное положение, т. е. к программе «Монитор».

Описание работы ИУК

Объект исследований размещен в специальном технологическом боксе (рис. 2). Первичные преобразователи физических величин и управляющие органы находятся в непосредственной близости и связываются с УСО комплекса через линии связи и линии контроля и управления. ИУК по рабочей программе КЛШ собирает и накапливает данные о ходе эксперимента в реальном масштабе времени, а также выдает сигналы управления экспериментом. Информация о ходе исследования поступает в пульттовую на матричный индикатор типа ПИУ-1. В ходе исследования измеряются следующие параметры: давление, температура, сила тока, ли-

Основные технические характеристики ИУК МСУВТ В7

Количество процессоров	3
Тип применяемых МП	КР580ИК80А
Объем адресуемой памяти (ОЗУ и ПЗУ), Кбайт	96
Время наработки на отказ, ч	10 000
Быстродействие МП, опер./с (типа пересылки регистр-регистр)	500 000
Число информационных каналов обмена с объектом:	
цифровых	2
аналоговых однопроводных	64
аналоговых двухпроводных	32
Основная приведенная погрешность АЦП-преобразования	0,2
Время АЦП-преобразования для плат:	
АВВ01, мкс	270
АВ08	10 мс — 2,56 с
Разрядность АЦП-преобразования, двоичн. разр. для плат	
АВВ01	10
АВ08	10—16
Время срабатывания входного элтрона канала цифрового ввода, мкс	2
Число ЦАП	4
Язык программирования	Макроассемблер, PL/M, БЕИ-СИК
Операционная система	ISIS-2, CP/M 80
Программное обеспечение	Монитор, редактор текстов, тестовые программы, библиотека прикладных программ, рабочие программы КЛШ
Средства дистанционной связи с другими ЭВМ	3-проводная ДМ. Физическая среда ДМ — коаксиальный кабель типа РК75

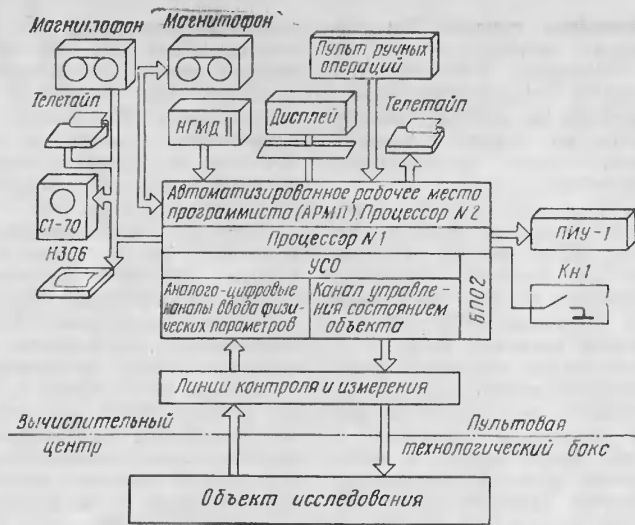


Рис. 2. Структурная схема управления процессом исследования с помощью ИУК

нейные перемещения. Окончательные результаты эксперимента записываются на НМЛ или НМД, где создается библиотека исследований. На печать телетайпа и на дисплей выдаются формализованный протокол исследований и полные данные по всем каналам в милливольтгах и физических единицах (в виде таблиц и графиков).

Программа KLSH

При подготовке программы KLSH приходилось решать противоречивую

задачу: ИУК должен работать по достаточно сложным алгоритмам в реальном масштабе времени, и в то же время управление им должно быть достаточно простым и рассчитанным на операторов, не имеющих фундаментальных знаний в области вычислительной техники и программирования.

Поэтому в основу разработки ПО был положен принцип: сложность выполняемых функций при простоте управления комплексом. Клавиатура телетайпа или дисплея используется

лишь в начале и конце эксперимента для записи в память служебных данных, а управляется комплекс одной кнопкой (К1, рис. 1, 2), расположенной в пультовой (25 м от ИУК).

После очередного нажатия происходит переинициализация программируемого контроллера прерываний. Таким образом, комплексом управлять проще, чем комплектом серийных приборов со множеством кнопок, тумблеров и переключателей, каждый из которых выполнял «жестко» определенные функции. Для успешной работы на комплексе достаточно знакомства с блок-схемой алгоритма KLSH (рис. 3).

После запуска программы на экране дисплея появляется текстовое сообщение: KLSH.NN, где NN — номер соответствующей версии.

1-й этап. Оператор заносит в память ЭВМ служебные данные: номера протокола и эксперимента, дату испытаний, номера опрашиваемых каналов и необходимый период опроса от 1 мс до 99,999 с. При работе по отлаженному протоколу данные пункты можно обойти, набрав символ «У». После программирования работы ИУК управление передается в пультовую.

2-й этап. После первого нажатия кнопки К1 проверяется работа таймеров, коммутаторов, преобразователей сигналов и первичных преобразователей, состояние помещения технологического бокса (необходимые блокировки в соответствии с требованиями техники безопасности). Пос-

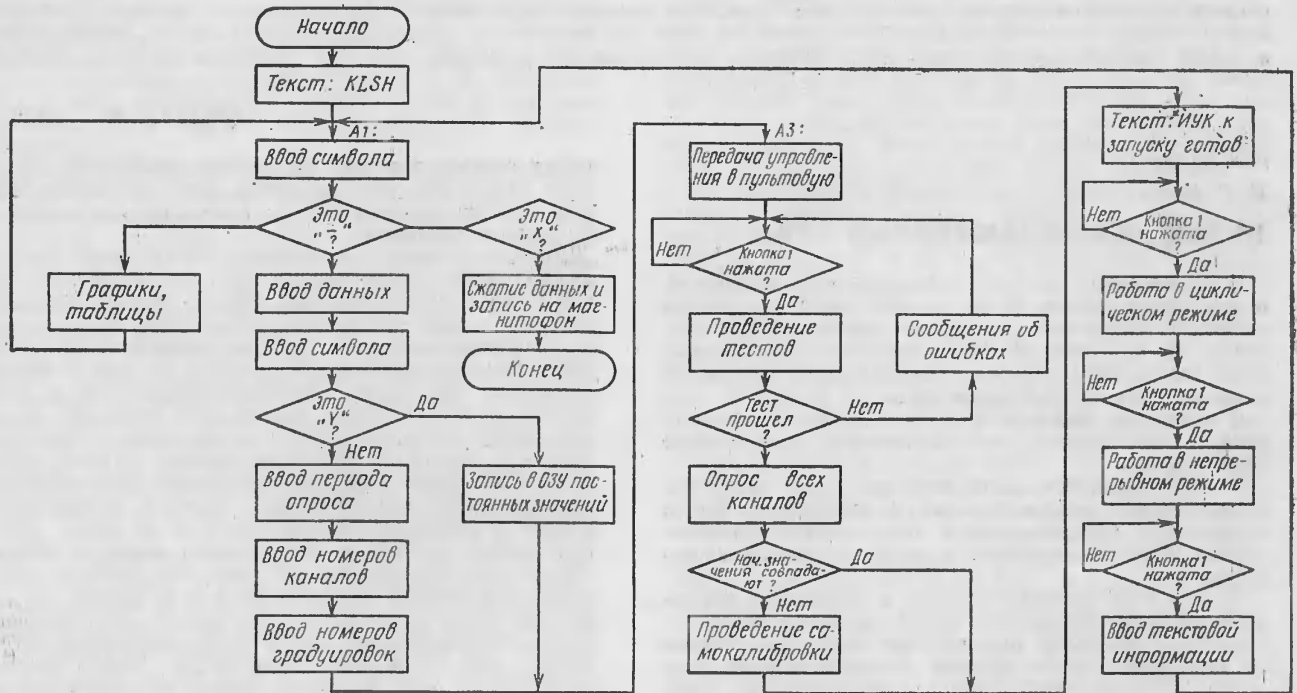


Рис. 3. Принципиальная схема KLSH

ле этого каждый канал десятикратно опрашивается и калибруется (если необходимо). Если в ходе диагностики обнаружена ошибка, соответствующее сообщение выводится на телетайп и индикаторный дисплей ПИУ-1 и происходит возврат к метке АЗ. Если все условия соответствуют заданным, то появляется сообщение «ИУК К ЗАПУСКУ ГОТОВ».

3-й этап. При следующем нажатии кнопки К1 комплекс работает в циклическом режиме. Записываются начальные значения параметров по всем опрашиваемым каналам.

4-й этап. При работе комплекса в непрерывном режиме с заданной частотой опроса результаты измерений записываются в ОЗУ. База данных организована так, что после опроса очередной группы каналов записывается и значение текущего времени измерения («привязка» к реальному времени).

5-й этап. После очередного нажатия кнопки К1 можно занести в электронную память необходимые служебные записи о ходе эксперимента и поведении исследуемого объекта, после чего программа возвращается в исходное состояние (А1). Полученные данные можно обработать и вывести на печатающее устройство телетайпа в удобном для исследователя виде в реальных физических величинах и (по необходимости) в единицах напряжения с указанием реального времени и времени с момента начала эксперимента. Однако более удобно представление результатов в виде графиков на графопостроителе Н-306. Кроме того, кривые изменения одного из измеряемых параметров можно наблюдать на экране осциллографа С1-70, а также можно проанализировать

форму изменения сигнала без дополнительных приборов — лишь с помощью телетайпа РТА 80. При этом по оси Х откладывается измеренное значение по данному каналу и печатается его номер, а каждая последующая строка соответствует новому циклу измерений.

Получив необходимые результаты, пользователь набирает символ «Х» (см. рис. 3), ИУК записывает информацию на внешний носитель. После окончания работы управление передается программе «Монитор».

Перспективы развития ИУК

Блочная-модульная структура комплекса и машинно-независимый интерфейс позволяют наращивать вычислительную мощность комплекса и увеличивать частоту опроса и число опрашиваемых каналов. Например, замена платы ПМВ01 на плату ПМВ02 (на основе 16-разрядного МП КМ1810ВМ86) увеличит ОЗУ и ПЗУ до 512 Кбайт, объем адресуемой памяти до 1 Мбайт, а быстродействие в 3—5 раз по сравнению с КР580ИК80А. В связи с тем, что затраты на ПО МПС непрерывно возрастают, основное внимание будет уделено развитию ПО. Благодаря использованию принципов структурного программирования при создании базовой программы KLSH очередные версии создаются в максимально короткое время. Используется основной принцип ГСП (государственной системы приборов), а именно: сложные устройства создаются из более простых унифицированных изделий, только аппаратное решение проблемы заменялось программным. При этом возможно взаимонезависимое совершенствование отдельных частей изделий без полного их обновления, использование различных

сочетаний модулей. Если в ранее разрабатываемых изделиях целью компоновки была минимизация аппаратуры, то для МСУВТ В7, широко использующих БИС, обеспечивающих непрерывное снижение стоимости операций и хранение информации, главный критерий компоновки — облегчение разработки ПО.

С помощью дистанционной магистрали ИУК будет связан с оборудованием ВЦКП (вычислительного центра коллективного пользования), а также с отдельными АРМ (автоматизированными рабочими местами) конструкторов и инженеров, организованными на базе современных микро- и мини-ЭВМ. Сети ЭВМ наиболее дешевы и целесообразны: они освободят исследователей от рутинной работы, сделают доступным информационные базы данных, позволят обеспечить переход к безбумажной технологии ведения работ и делопроизводства.

Телефон 189-52-86, Москва.

ЛИТЕРАТУРА

1. Новопашенный Г. Н. Информационно-измерительные системы. — М.: Высшая школа, 1977.
 2. Шереметьевский Н. Н., Долкарт В. Н. Магистрально-модульные микросредства управляющей вычислительной техники // Микропроцессорные средства и системы. — 1984. — № 2. — С. 24—27.
 3. Рокленд К., Эволт, Бен Л. Ги. Локальная сеть — комплекс микропроцессоров, превосходящая крупные ЭВМ по производительности // Электроника. — 1983. — № 24.
 4. Стивен У. Филдс. Личный компьютер становится рабочим инструментом инженера: Обзор // Электроника. — 1983. — № 24. — С. 91.
- Статья поступила 10.12.87

ФОРУМ „МП“

УДК 681.3.06

И. С. Лосев

БУДУЩЕЕ КОМПЬЮТЕРНЫХ ИГР

Компьютерные игры. Их показывают по телевизору, о них пишут статьи. И все больше детей и взрослых получают возможность не только слушать о них рассказы, но и играть. По мере того, как персональные компьютеры (ПК) становятся реальностью, становятся реальностью и компьютерные игры.

И возникает проблема: как относиться к распространению компьютерных игр? Существуют две крайние точки зрения.

Первая объявляет компьютерные игры средством обольщения детей и взрослых, а деятельность по их созданию и распространению приравнивает к развращению несовершеннолетних и распространению наркотиков.

Вторая точка зрения призывает к всеобщей и безоговорочной компьютеризации досуга, основным средством которой объявляются компьютерные игры. Представители этой точки зрения приняли постановление об организации 200 досуговых центров, оснащенных ПК. Заметьте, что, если необходимо срочно создавать сотни иг-

ровых центров в стране, то, видимо, заниматься созданием самих игр уже не нужно (они предполагаются уже существующими), так что в этом вопросе противоположности сходятся.

Другими словами, сторонники обеих точек зрения рассматривают компьютерные игры как достаточно законченное и зрелое явление, которое необходимо либо запрещать, либо пропагандировать. Точка зрения автора, напротив, состоит в том, что компьютерные игры, обеспечивающие сегодня, как правило, жалкое, а иногда уродливое или вредное времяпрепровождение, в действительности — зародыш нового вида искусства, которое займет достойное место в будущем, образовав вместе с кино и телевидением триаду искусств, рожденных XX веком. Сегодняшний уровень компьютерных игр примерно соответствует уровню, достигнутому в кино в короткометражках 1912 г. В то время большая часть критиков с негодованием отвергла дикую идею о том, что кино сможет иметь какое-нибудь отношение к искусству. «Кинематограф и искусство — явление различного порядка. ...Может ли кинематограф стать самостоятельным искусством? Разумеется, нет». Автор этих строк — не какой-нибудь заплесневелый ретроград, а один из самых смелых новаторов искусства начала XX века — Владимир Маяковский,

На чем же, кроме таких поверхностных аналогий, основана высказанная выше надежда? Дело в том, что так называемые зрелищные искусства (как древние — театр, так и новые — телевидение, кино) всегда стремились возможно полнее реализовать непосредственное участие зрителя в предлагаемом зрелище. В театре попытки сделать зрителя участниками представления были особенно активны в первой четверти века в России. Что касается кино и телевидения, то по техническим причинам, очевидно, возможна лишь иллюзия участия зрителя, однако эта иллюзия современными искусствоведами объявляется одним из основных художественных принципов кино и главнейшей отличительной особенностью телевидения. М. Ромм пишет: «Разница между восприятием зрелища в театре и восприятием зрелища с экрана заключается, в частности, в том, что в театре зритель неподвижен и непрерывно смотрит как бы общим планом события, развернутые на него. А в кинематографе зритель как бы подвижен, он как бы поднимается вместе с аппаратом на сцену, входит внутрь мизансцены, рассматривает ее то сверху, то снизу, подходит то к одному актеру, то к другому... Таким образом, зритель кинематографа активно участвует в жизни» [1].

В компьютерной игре развитие сюжета определяется (в рамках, заданных автором игры) действиями играющего и, таким образом, он реально, а не иллюзорно участвует в развивающемся действии. Поэтому впервые в истории искусств появляется техническая возможность создавать произведения, которые будут не пассивно восприниматься зрителем, а активно и творчески создаваться им. Невероятно, чтобы такая возможность не оказалась рано или поздно реализованной.

Ниже мы обсудим, какие проблемы возникают в конструировании компьютерных игр. Но поскольку сегодня лишь незначительная часть читателей пользуется дома или хотя бы у себя на работе ПК, для которого существуют современные игровые программы, то перед этим опишем примеры компьютерных игр.

Компьютерные игры разделяются на довольно большое число различных жанров, иногда взаимопроникающих, иногда мало похожих друг на друга. Нас здесь более всего интересуют так называемые игры типа «Приключения» (первая игра этого типа, как и фирма, выпускавшая эту игру и ее многочисленных потомков, называлась «Приключенис»). Об играх других жанров можно прочитать в интересной статье Г. Б. Кочеткова [2], а здесь мы приведем описание одной из игр типа «Приключения», взятое из американского журнала «Байт» [3].

«Рассказ начинается с того, что неопытный молодой парень получает торговый космический корабль, некоторую сумму денег в банке и контракт на доставку большого количества товаров в новую космическую колонию. Корабль в ужасном состоянии. Его владелец никогда раньше не пилотировал космический корабль и ничего не знает о космической торговле. Само собой разумеется, денег не хватает на то, чтобы починить и снабдить топливом корабль и выполнить контракт... Текст правил сообщает вам это и сообщает некоторую техническую информацию об управлении игрой; все остальное — ваше дело. Это вполне соответствует сценарию. Предполагается, что вы будете экспериментировать, чтобы узнать больше. Вы можете вывести своего героя из космического корабля в город, где вы можете заходить в магазины за запчастями, в бары за едой, пивом и информацией, на рынки и в другие места... Единственный способ узнать, что внутри здания, — это войти в него. В некоторые здания войти нельзя, но вы не узнаете об этом, пока не попробуете. Другие здания очень важны, но вы не знаете, что это такое, пока не войдете внутрь: снаружи нет вывесок. Войдя внутрь, вы можете общаться с персонажами, которых там встретите, но вы не можете задать им тот вопрос, который вас интересует, и должны ждать,

не сообщат ли они то, что вы хотите. Дело в том, что не существует способа сказать что бы то ни было, кроме того, что можно выбрать из появляющихся время от времени списков фраз. Например, история, рассказанная в тексте правил, заставляет вас думать, что сведения о товарах для торговли вы узнаете в барах, но единственно, с кем я там встречался, были люди, предлагавшие мне то, что оказывалось наркотиками».

Из описания не ясно, как изображается на экране компьютера город, космический корабль и другие места действия, но в действительности это — рисованная мультипликация, так что игра напоминает рисованный мультипликационный фильм, только развертывающийся под управлением играющего. Действительно, все, что зритель увидит на экране, должно быть предварительно некоторым способом заложено в ЭВМ, и этот процесс имеет много общего с деятельностью художника-мультипликатора. В конструирование компьютерной игры входит и деятельность, аналогичная деятельности художника-мультипликатора, и деятельность, которую следует признать режиссерской, и работа драматурга или писателя. Далее мы опишем специфику всех этих деятельностей, начав с наиболее понятной — создания рисунков внутри ЭВМ.

Заметим, что в процессе развития мультипликационного кино создавались и создаются различные технологии мультипликации, использующие различные технические средства. Аналогично этому новые все более совершенные компьютеры имеют все более совершенную специализированную технику для легкого создания все более сложных мультипликаций.

Пример такой техники — «спрайтовая» графика, во многом похожая на классическую технологию рисованной мультипликации. В ней программист должен сначала нарисовать неподвижный фон. Далее движущиеся фигуры разбиваются на квадраты относительно небольшого размера. Рисунок на каждом таком квадрате делается отдельно (такой квадрат вместе с рисунком на нем называется спрайтом, что в переводе означает «эльф»).

Все точки спрайта движутся одновременно, как будто рисунок нанесен на отдельном прозрачном листике. Каждый листик помещен на одну из нескольких плоскостей, перпендикулярных лучу зрения, так что окрашенные части листиков, лежащих ближе к зрителю, заслоняют листики, находящиеся непосредственно за ними (неподвижный фон помещен на самом дальнем листике). Спрайт может быть только одного цвета, но зато цвет всех точек спрайта может быть практически мгновенно изменен на любой другой имеющийся в системе цвет. Спрайт даже может быть сделан прозрачным.

Из приведенного описания нельзя понять, каким способом программист наносит рисунок на спрайт или рисует фон. Но если наличие самой возможности работать со спрайтами и различные технические ограничения (количество цветов на экране, разрешение, количество спрайтов и т. д.) определяются аппаратурой, т. е. маркой компьютера, то способ задания рисунка, как и способ задания его движения, можно сделать более или менее удобным для автора игры, если ранее для этой цели была написана специальная программа (редактор мультипликации). Без такой программы процесс задания рисунка весьма неудобен и сводится к набору длинных списков чисел. (Этот пример иллюстрирует, насколько разные вещи «умет ЭВМ» для людей трех разных категорий: системного программиста, создающего редактор мультипликации, прикладного программиста, рисующего с помощью этого редактора декорации в создаваемой им игре, и конечного пользователя, играющего в эту игру с ЭВМ).

Описанная здесь техника (она в тех или иных вариантах существует на многих западных домашних компьютерах) дает по крайней мере принципиальную возможность создавать рисованную мультипликацию в компьютерной игре, не отличимую от реальной съем-

ки, однако необходимая для этого вычислительная мощность очень велика. Голливудские фирмы, используя синтезированное компьютером изображение при съемке коммерческих фильмов (это даст возможность получать замечательные трюковые эффекты), вынуждены закупать для этого самые мощные суперЭВМ за десятки миллионов долларов.

В обозримое время ПК не будут обладать соответствующей вычислительной мощностью. Но это и не нужно. Видимо, адекватным техническим средством для реализации компьютерных игр высокого уровня будет мощный ПК, снабженный оптическим диском. На выпускаемых сегодня оптических дисках можно записать 45-минутный цветной телевизионный фильм. При этом, во-первых, компьютер может имеющиеся там десятки тысячи кадров показывать в порядке, задаваемом программой и определяемом действиями пользователя, и во-вторых, накладывать на эти кадры заранее синтезированные изображения отдельных предметов или фигур. Конечно, такая возможность менее интересна, чем синтезирование изображения внутри компьютера, поскольку можно управлять лишь очень небольшим числом заранее синтезированных объектов и последовательностью готовых кадров. Такого рода игры — дело будущего, во всяком случае автору не известно ни одной компьютерной игры, использующей оптический диск. Впрочем, если бы сегодня удавалось создавать компьютерные игры на уровне современной мультипликации, то это было бы колоссальным шагом вперед.

Наиболее сложная задача — организовать взаимодействие между Героем (персонажем игры, управляемым игроком), внешним миром и другими персонажами игры. При этом сложность вообще не связана с программированием. Для объяснения этой сложности ползла аналогия с кино.

В кино существуют два типа сценариев: литературный, где говорится о том, что происходит, общими словами (почти так же, как в пьесе или рассказе), и режиссерский, где этот рассказ раскадрован, разбит на кадры, т. е. на отдельные сцены с указаниями, каким способом они должны быть сняты (монтажные планы и т. д.). По режиссерскому сценарию снимается материал, который в процессе монтажа становится фильмом.

Компьютерная игра может начинаться с литературного сценария, рассказывающего об игре и ее типичных ситуациях и идеях (часть этого сценария может войти в текст правил игры), режиссерского сценария, возможно, более точно описывающего поведение персонажей игры, и собственно программной игры, реализующей этот сценарий. Режиссерский сценарий может быть текстом на русском языке, а вовсе не программой, и все-таки в нем содержится основная специфика компьютерной игры.

Дело в том, что режиссер в кино или театре описывает то, что должно быть на экране, а автор компьютерной игры — лишь то, что может быть (реализуется оно или нет, будет зависеть от действий играющего).

Попробуем пояснить это на примере. Предположим, что нас интересует эпизод, в котором Герой стреляет из лука в чудовище. В литературном сценарии ему может соответствовать одна фраза: «Метким выстрелом Герой сшибает чудовище с ног». В кино соответствующий фрагмент режиссерского сценария может выглядеть так.

1. Герой натягивает лук, целится и спускает тетиву (средний план).

2. Стрела пересекает разделяющий Героя и чудовище ров и достигает чудовища (общий план).

3. Мы видим, как стрела вонзается в чудовище и оно падает (средний план).

Опишем фрагмент правил поведения персонажей компьютерной игры, обеспечивающих возможность увидеть на экране компьютера тот эпизод, который мы увидим в кино. В этом эпизоде участвуют три персона-

жа: Герой, стрела и чудовище. В правилах поведения Героя имеется очень много главок, соответствующих действиям, которые играющий может попросить выполнить Героя. Нас интересует главка Стрелять. Ниже — фразы в скобках — это комментарии, которые в настоящем тексте правил отсутствовали бы.

Герой

Стрелять

Если в руках Героя нет лука, то скажи играющему: «Не могу стрелять без лука». Если в руках Героя есть лук, то:

1. Подними лук. (Заметим, что этот пункт, как и предыдущий, можно рассматривать в качестве приказа еще одному персонажу — Экрану. Получив это приказание, он должен передвинуть спрайт, на котором нарисован лук.)

2. Пошли Стреле приказание: «Выстрел из точки X по направлению В с силой А. (Здесь X — это точка, где в момент выстрела находился Герой, а В — направление его лука. Сила — это одна из внутренних характеристик Героя, которая меняется при тех или иных его действиях, скажем, когда он выпивает волшебный напиток или получает рану. Правила этого изменения перечислены в других главках, связанных с Героем.)

Правила действий Стрелы состоят из единственной главки Выстрел.

Стрела

Выстрел (из точки X по направлению В с силой А)

1. Найди первое препятствие стреле при выстреле из точки X по направлению В.

2. Проведи по экрану изображение стрелы от точки X до препятствия, после чего убери его с экрана.

3. Если препятствие — это чудовище, то пошли ему сообщение: «Выстрел с силой А».

Правила действий чудовища состоят из большого числа главок. Нас здесь интересует только главка Выстрел.

Чудовище

Выстрел (по направлению В с силой А)

Если сила выстрела А меньше прочности шкуры (это одна из внутренних характеристик чудовища), то пошли себе сообщение: «Прыгни» (навстречу В).

2. Если сила выстрела А больше прочности шкуры, то вместо изображения стоящего чудовища помести изображение лежащего.

Мы не будем здесь выписывать даже содержание главки «Прыгни» (относительно чудовища), без которой этот маленький фрагмент правил неполон. Надеюсь, у читателя создается некоторое впечатление о сходствах и различиях режиссерских сценариев в кино и в компьютерных играх.

Заметим, что записанная нами система правил в компьютерной игре не эквивалентна приведенному выше режиссерскому сценарию в кино, поскольку то, что видит зритель в игре, соответствует одному общему плану на протяжении всего эпизода (программист обычно выражает это так: на протяжении эпизода не происходило смены экранов), а не тем трем разным монтажным планам, из которых должен быть смонтирован эпизод согласно режиссерскому сценарию в кино.

Однако, как и в кино, в сложных играх необходимо переходить от одного экрана к другому — эквивалент кинематографического монтажа. Монтаж в компьютерной игре всегда будет отличаться от кинематографического тем, что смена планов будет происходить не по воле режиссера, а в результате действий играющего. Вообще говоря, даже правила действий персонажей могут меняться от экрана к экрану. Таким образом, действия, приводящие к смене экрана, связывают отдельные сцены в единый сюжет.

Как же может строиться сюжет в компьютерной игре? В идеале ее литературный сценарий должен содержать множество альтернативных сюжетов.

Параллельные сюжеты сами по себе время от времени встречаются в классической литературе. Прежде всего упомянем сказки, где показывается, как действия, выбираемые падчерницей, приводят ее к выигрышу, в то время как действия, выбираемые в тех же самых обстоятельствах родной дочкой, приводят к проигрышу. Для сказки такая структура сюжета дает возможность с максимальной легкостью реализовывать воспитательную функцию и компьютерная игра разделяет эту легкость. Параллельные сюжеты часто встречаются в современной фантастике в контексте путешествий во времени. Однако в отличие от сказок в этих произведениях часто показывается, что отклонение от первоначального сюжета, вызванное принятием других решений, не приводит к другому конечному результату. Из исследующих эту тему произведений «серьезной литературы» надо упомянуть пьесу Дюрренмата «Испытание», где некий таинственный посредник дает возможность герою несколько раз возвращать время назад и пробовать другие варианты поведения в его жизни — результат удручающе одинаков. Но во всех этих примерах сюжеты действительно параллельны: после того, как один выбор сделан, каждый из сюжетов развивается уже практически без вариантов, а из одного сюжета нельзя вернуться к другому. Компьютерная игра может и, вероятно, в идеале должна обеспечивать гораздо более богатое взаимодействие сюжетов.

Автор хотел бы, чтобы все высказанные выше соображения иллюстрировали главный тезис — компьютерные игры обладают своими специфическими средствами выражения, потенциально позволяющими решать новые художественные задачи.

Что же нужно для появления компьютерных игр, где эти задачи ставились и решались? Прежде всего возможно более полное отделение процесса создания игры от программирования. Только тогда создание игр сможет стать профессиональной деятельностью. Сегодня значительная часть коммерческих игр создается на ассемблере конкретной машины. Это сужает круг их потенциальных создателей до предела, исключает возможность перенесения игры на ЭВМ другого типа, и ограничивает сложность программы. Игра должна формулироваться не на языке, удобном для ЭВМ, а на языке, удобном для выражения игры.

К счастью, некоторый прогресс в этом направлении происходит. Выше уже упоминались редакторы мультипликации, избавляющие от программирования создателя «зрительного образа» игры. За последний год в Советском Союзе появились по крайней мере две такие разработки [4]. Более сложная задача перевода режиссерского сценария компьютерной игры (совокупности правил действий персонажей) в окончательную компьютерную программу. Но и здесь в последнее время прогресс существен. Правила действия Героя, стрелы и чудовища, приведенные выше, могут показаться очень далекими от реальной программы не только новичку, но и опытному программисту, привыкшему программировать на Фортране, БЕЙСИКЕ или Паскале. В действительности же эти правила действий можно рассматривать практически как готовую запись программы (точнее, работа по переводу их в программу является простой и чисто технической). Только программа эта будет написана не на одном из перечисленных языков программирования, а на каком-нибудь объектно-ориентированном языке. Объектно-ориентированные языки — это новый тип языков, созданный в конце 70-х — начале 80-х годов [5]. В них программа состоит из совокупности объектов (персонажей), обменивающихся сообщениями. Объектно-ориентированные языки становятся все более популярными на Западе, но пока еще практически отсутствуют у нас [6]. Для создания игр необходимо иметь объ-

ектно-ориентированный язык со встроенными в него средствами создания мультипликации.

Реализация такого языка и других средств программного обеспечения, необходимых для освобождения создателя игры от скучных и не нужных деталей, довольно трудоемка. Почему же нужно ею заниматься? Почему вообще нужно заниматься созданием компьютерных игр?

Даже если считать, что шансы на то, что компьютерные игры станут рано или поздно искусством, не очень велики (сам автор считает, что они близки к 100%), то награда — новое искусство — так велика, что стоит рискнуть. Но есть и две более прагматические причины.

Компьютерные игры могут быть практически полезны как очень важный новый метод обучения. В этой статье нет места для подробного обсуждения этих вопросов, но автор считает, что уже сегодня можно создавать игры, не менее занимательные, чем имеющиеся в коммерческой продаже на Западе, но отличающиеся тем, что в процессе их освоения и игры школьник будет вынужден изучить некоторый раздел школьной программы.

Наконец, пожалуй, наиболее важной сегодня причиной является та, что сегодняшний средний уровень компьютерных игр вызывает страх: дело не в том, что дети (да и взрослые) тратят на них время, не обучаясь ничему. Ситуация намного хуже. Сегодняшние компьютерные игры, пришедшие к нам с запада, эффективно действуют на играющих детей, внедряя в их подсознание свою идеологию. И для большей части западных игр, циркулирующих в Советском Союзе, эту идеологию можно выразить словами «Убей их всех!». И если мы не хотим через некоторое время оказаться перед проблемой нового духовного наркотика, то с такими играми надо бороться. Но бороться не запретительными мерами, поскольку практически запреты лишь прячут проблему под ковер, а создавая более интересные игры с другой идеологией.

Телефон 297-34-76, Москва

(Окончание статьи см. на с. 94)

УДК 681.03

В. В. Липаев

НЕКОТОРЫЕ ОСОБЕННОСТИ ПОДГОТОВКИ СПЕЦИАЛИСТОВ ПО ПРОЕКТИРОВАНИЮ ПРОГРАММНЫХ СРЕДСТВ

Специфика программных средств для встроенных ЭВМ

К ряду типов программ, в частности для встроенных ЭВМ, непосредственно управляющих объектами, роботами, технологическими процессами, гибкими автоматизированными производствами в реальном масштабе времени, предъявляются особые требования. Такие программы обладают наибольшей экономической эффективностью при использовании в народном хозяйстве, их роль непрерывно возрастает, они квалифицируются как программные средства и продукция производственно-технического назначения. Программы как промышленные изделия снабжаются комплектами документации, обеспечивающими их производство, освоение и квалифицированную эксплуатацию.

Принципиальная особенность программных средств (ПС) — коллективность разработки, производства и применения, с разделением труда в зависимости от квалификации и специализации. При разработке ПС сохраняются все проблемы, характерные для проектирования, производства и применения сложных промышленных изделий и систем. Поэтому технологию их созда-

ния правильнее называть технологией проектирования программных средств, а не технологией программирования, которая является только частью процесса создания программ. Программные средства должны характеризоваться измеримыми техническими показателями количества и качества, а также экономическими (производительностью труда, распределением затрат и т. д.). Эти характеристики ПС должны обеспечивать планирование и управление созданием и применением ПС в масштабах страны, отдельных отраслей, предприятий и проектов. Процесс проектирования ПС в основном аналогичен современному автоматизированному проектированию сложных изделий в радиоэлектронике и машиностроении.

Следует подчеркнуть особую опасность последствий неправильного функционирования ПС в системах реального времени. Качество и надежность ПС определяют не только эффективность технологических процессов, но и величину риска возможных аварий с огромным материальным ущербом и даже гибелью людей. Ошибки в программах, связанных с проектированием изделий, строительством и экономикой, также могут приводить к большому материальному ущербу, хотя и менее связанному с риском для жизни и здоровья людей. Поэтому большое значение приобретают контролируемый технологический процесс создания и испытаний ПС с тщательным моделированием внешней среды. При разработке таких сложных комплексов программ необходим четко организованный труд больших коллективов специалистов. Их труд должен быть автоматизирован и регламентирован системой стандартов и инструкций на все этапы и компоненты технологического процесса, а также на качество компонентов и изделия в целом.

Требования к разработчикам программных средств

Создание программных средств с перечисленными свойствами значительно отличается от разработки и применения небольших программ для частных вычислений. Не отрицая важности массовой первичной грамотности в области программирования, данная работа обращает внимание на необходимость глубокого профессионализма при проектировании ПС и превращения его в строгую инженерную дисциплину.

Вследствие этого имеются особые требования к подготовке специалистов по всем этапам жизненного цикла ПС встроенных ЭВМ: недостаточно навыков программирования, необходимы глубокие знания системной техники и технологии проектирования сложных комплексов программ. Технология создания программной продукции, крупномасштабное проектирование ПС, разделение труда при разработке комплексов программ, организация коллективов и экономики таких разработок должны стать важной частью подготовки специалистов по ПС в технических вузах. Около 15 лет назад очень точно сформулировал эти требования академик А. П. Ершов [1]: «программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению, в сочетании с эдиссоновским талантом соорудить все, что угодно, из нуля и единицы. Он должен сочетать аккуратность бухгалтера с проинтерпретацией разведчика, фантазию автора детективных романов с трезвой практичностью экономиста. А кроме того, программист должен иметь вкус к коллективной работе, понимать интересы пользователя и многое другое...» (выделено автором).

Инженеры должны свободно владеть двумя-тремя языками программирования и процессом отладки программных компонентов. Важно с самого начала обучения готовить их к коллективному созданию множества программных компонентов (с профессиональным разделением труда), подлежащих объединению в сложные

комплексы программ. Обучение индивидуальной разработке программ не исключается, однако даже на начальных этапах подготовки необходимо прививать студентам системное мышление [2]. Подготовка в этом направлении должна поддерживаться циклом теоретических курсов и практических занятий по крупномасштабному проектированию сложных ПС; так же как для студентов, специализирующихся по вычислительной технике, есть курсы по общей теории и архитектуре ЭВМ и их комплексному проектированию.

С этих позиций для технических вузов разработан цикл профилирующих курсов по программному обеспечению для подготовки инженеров по специальности 0646. Общая программа подготовки инженеров значительно отличается от программы для университетов по специальности «Информатика и системное программирование» [4], хотя ряд курсов достаточно близок. Этот цикл начинается с курса «Технология проектирования комплексов программ управления», который разработан и с 1970 г. в течение 13 лет читался автором статьи, а затем его коллегой по кафедре. Материал курса [3, 5] за эти годы несколько модифицировался и его чтение постепенно перешло с пятого курса на третий. Тем самым роль его изменилась с объединяющей и завершающей цикл курсов по программному обеспечению АСУ на роль, подготавливающую и предшествующую курсам по проектированию программного обеспечения реального времени.

Цель курса — изучение методов структурного построения комплексов и компонентов программ управления в реальном масштабе времени, а также технологии, методов и средств автоматизации разработки таких комплексов. Курс рассчитан на 78 часов аудиторных занятий, 64 часа лекций и 14 часов практических занятий, содержит 16 тем (от трех до пяти лекционных часов). На пятом семестре изучается объект проектирования, его общие характеристики, методы структурного построения комплексов программ и их унифицируемые компоненты, распределение ограниченных ресурсов встроенных ЭВМ для их реализации. Шестой семестр посвящен изучению технологии и экономики разработки сложных комплексов программ встроенных ЭВМ, методов и средств автоматизации проектирования программ, их отладки и испытаний. На семинарских занятиях изучается структурное проектирование комплекса программ, распределение вычислительных ресурсов и расчет эффективности средств повышения надежности программ. Курс вдвое больше по объему по сравнению с аналогичным [4] и отличается большей практической направленностью. Кроме того, следует отметить, что он читается на год раньше (пятый и шестой семестр вместо восьмого).

Содержание курса «Технология проектирования комплексов программ управления»

Пятый семестр

1. Вводная лекция курса и цикла курсов «Программное обеспечение АСУ». Историческая справка о развитии технологии проектирования программ и связь со смежными дисциплинами. Комплексы программ управления как сложные системы. Примеры применения комплексов программ. Проблемы проектирования сложных программ — функциональные, структурные, технологические, эксплуатационные. Состав и функции общего программного средства.

2. Технологическая, методическая, инструментальная и организационная поддержка технологии проектирования программ. Понятие жизненного цикла программ. Этапы проектирования программ, основные задачи и принципы их решения, Уровни автоматизации техноло-

гии. Понятие резидентных и кросс-систем автоматизации.

3. Техничко-экономические критерии оценки технологий проектирования программ. Принципы оптимального распределения ресурсов в процессе проектирования комплексов управляющих программ. Основные зависимости показателей конструктивной эффективности программ от уровня автоматизации технологии и других параметров. Трудоемкость проектирования в зависимости от объема программ и степени автоматизации технологии. Методика выбора технологии и оценки ее экономической эффективности.

4. Принципы нисходящего и восходящего проектирования. Модуль-иерархическое построение комплексов программ и структур данных. Понятие и правила структурного программирования модулей. Программные спецификации, их структура и методы применения. Стандартизация оформления модулей. Принципы определения сложности программ и методы ее снижения. Оценка сложности модулей, межмодульных связей и структур данных.

5. Классификация управляющих систем по времени реакции и длительности решения задач, влияющие эти характеристики на структуру программ. Назначение и взаимодействие типовых компонентов операционной системы реального времени и системы функционального контроля в различных режимах. Типовая структура зон оперативной памяти. Структура и функции программ организации вычислительного процесса и функционального контроля. Назначение, функции и блок-схемы диспетчеров, таймеров, программ начального пуска, взаимодействие многомашинных комплексов, обмена с внешними абонентами, контроля загрузки.

6. Методы и эффективность распараллеливания программ. Управление ресурсами и распределение программ при параллельных вычислениях. Методы взаимодействия программ в многомашинных и многопроцессорных вычислительных системах в стационарных и переходных режимах. Алгоритмы организации параллельных вычислений и взаимодействия комплексированных ЭВМ.

7. Критерии и методы распределения вычислительных ресурсов. Статистические и динамические методы. Параметры, определяющие эффективность организации вычислительного процесса. Методы распределения ресурсов на основе расписаний, приоритетные методы, квантованное обслуживание. Методы оценки эффективности диспетчеризации по штрафам за ожидание обслуживания и за потерю сообщений, по изменению эквивалентной памяти и производительности вычислительной системы.

8. Характеристики эффективности дисциплин диспетчеризации в зависимости от основных параметров. Оптимальное число приоритетов. Вероятность потери сообщений при различных дисциплинах диспетчеризации и ограниченной буферной памяти. Расчет объема накопителей для приема и выдачи сообщений. Оптимизация распределения внешней памяти. Методика проектирования систем организации вычислительного процесса. Расчет эффективности распределения ресурсов.

Шестой семестр

9. Планирование разработки крупных комплексов программ. Исходные документы и ресурсы для проектирования. Состав эксплуатационных, технологических и исследовательских документов. Основы ЕСПД. Принципы организации коллективов для создания комплексов программ. Характеристики трудозатрат и потребностей в специалистах по этапам жизненного цикла программ. Сетевое планирование разработки программ. Укрупненный сетевой график создания комплексов программ. Сетевой график разработки технического проекта.

10. Функциональные задачи систем автоматизации технологии. Основные требования к ним и их взаимосвязь. Методы решения задач. Структурная схема и взаимодействие компонентов современной системы автоматизации

технологии проектирования комплексов программ. База данных проектирования. Подготовка технологических систем к условиям конкретной разработки.

11. Система языков — командный, спецификаций, программирования, отладки. Особенности языков программирования для задач управления. Унификация записи текстов программ и интерфейса модулей. Описание глобальных переменных и базы данных проекта. Паспорта программ. Трансляция, загрузка, редактирование связей. Комплексирование модулей. Контроль текстов программ.

12. Понятие эталона и ошибки в программе. Корректность программ. Математические модели описания ошибок. Основные статистические характеристики ошибок. Принципы оценки отлаженности программ. Методы определения рациональной длительности отладки. Формальные методы доказательства корректности программ.

13. Основные понятия надежности программ. Факторы, определяющие характеристики надежности. Методы оперативного обнаружения искажений вычислительного процесса, программ и данных; программной защиты от отказов и сбоев при исполнении программ. Эффективность оперативного рестарта программ. Оптимизация затрат на обеспечения надежности. Методика проектирования комплекса программ с заданной надежностью.

14. Этапы отладки комплексов программ и методы автоматизации отладки. Принципы отладки сверху-вниз и снизу-вверх. Графовая модель программы. Структурный контроль модуля. Методы планирования автономной отладки программных модулей и подготовки тестов. Язык автономной отладки и отладочное задание. Методы объединения отладочного задания и отлаживаемой программы. Исполнение отлаживаемых программ — компиляция и интерпретация. Схемы систем автоматизации отладки и корректировки программ. Пакетный и диалоговый режим отладки. База данных тестов и контроль уровня отлаженности.

15. Статическая отладка комплексов программ. Комплексирование модулей и контроль корректности связей. Динамическая отладка в реальном времени. Принципы имитации внешней среды и реального времени. Методы определения пропускной способности комплекса программ и его реализуемости в реальном времени. Схемы построения имитационно-моделирующих стендов для отладки.

16. Организация испытаний комплексов программ. Основные показатели качества. Исходные и отчетные документы при испытаниях. Особенности испытаний на надежность. Задачи и методы сопровождения комплексов программ. Версии программ — иерархия выполнения корректировок. Методы защиты от их разрушения. Тиражирования версий и контроль проведения изменений. Оформление прекращения сопровождения.

ЛИТЕРАТУРА

1. Ершов А. П. О некоторых человеческих и эстетических факторах в проектировании // Кибернетика. — 1972. — № 5. — С. 3—8.
2. Хоар Ч. А. Р. Программирование как инженерная профессия // Микропроцессорные средства и системы. — 1984. — № 4. — С. 53—60.
3. Липаев В. В. Проектирование математического обеспечения АСУ. — М.: Сов. радио, 1977. — 400 с.
4. Лавров С. С., Слисенко А. О., Цейтин Г. С. Проект план-программы по специальности «Информатика и системное программирование» // Микропроцессорные средства и системы. — 1985. — № 4. — С. 20—28.
5. Технология проектирования комплексов программ АСУ / В. В. Липаев, Л. А. Серебровский, П. Г. Гаганов и др.; Под ред. Ю. В. Асафьева и В. В. Липаева. — М.: Радио и связь, 1983. — 264 с.

Статья поступила 17.02.87

РАБОТЫ ПО СОЗДАНИЮ ЗАКОНОДАТЕЛЬСТВА ИНФОРМАТИКИ В СССР

Происходящие в структуре производительных сил революционные перемены все более заметно воздействуют на производственные отношения — появляются специализированные предприятия по производству программных средств [1], развивается рынок баз данных [2], знания включаются в товарооборот, а затем начинают доминировать как основной источник национального богатства. Неуклонное возрастание роли информационных процессов вызывает ломку правоотношений, сложившихся в период развития вещественно-энергетического производства.

В связи с выделением вычислительных центров как специализированных органов по переработке машинной информации (МИ) общественные отношения по поводу сбора, хранения, обработки, передачи и отображения информации с использованием ЭВМ также обособляются. Такие общественные отношения характеризуются появлением нового предмета правового регулирования, который в последние годы дополнился специфическим методом рассмотрения (метод двух интеллектов), заключающимся в учете растущих «интеллектуальных» способностей вычислительных систем (ВС).

В настоящее время в основном произошло отделение труда по переработке МИ от иных видов труда, осуществился первый технологический этап — переход к товарному производству МИ специально для чужого потребления. Это требует установления правового статуса МИ в экономических связях и юридического оформления взаимоотношений участников соответствующих общественных отношений. Развитие интеллекта ВС потребовало рассмотрения их в качестве действия и позволило сделать вывод (пусть и не бесспорный) о целесообразности введения интеллектуальных ВС в правовую систему в качестве принципиально нового фактивного субъекта* [4].

Специфика предмета и метода правового регулирования общественных отношений в сфере информатики стала основой для формирования новой комплексной отрасли права, отражающей гражданские, государственные, административные, уголовно-правовые и процессуальные общественные отношения. Особенности информационно-правовых отношений, предмета и метода правового регулирования весьма существенно влияют на уже сложившуюся законодательную систему [5—7]. Искусственная рабочая сила в виде ВС порождает самостоятельный экономический режим информационных ресурсов, требует пересмотра понятий «стоимость» и «цена», вводит информационно-вычислительные ресурсы в круг балансируемых факторов развития народного хозяйства [8]. Наконец, специализация производителей МИ вызывает дробление ранее единых понятий: информационные ресурсы, программирование и т. п. При этом оформляется структура индустрии информатики, обособляются органы управления различными сферами этой индустрии, развиваются специфические методы управления.

Расширение масштабов и значимости информационных общественных отношений, нарастание противоречий в производственной и социальной сферах сделали настоятельно необходимой постановку проблемы создания законодательства информатики [9] и развертывание в СССР соответствующих работ. В составе научно-технического совета ГКВТИ СССР образована секция нормативно-правового обеспечения информатики. Предприя-

тия ГКВТИ СССР совместно с институтами АН СССР, Минюста СССР, ведущими вузами начали целенаправленную работу по созданию нормативного массива, призванного послужить базовой основой законодательства.

Безвозмездное распространение МИ противоречит общественным интересам — вот центральная экономическая посылка формируемого правового механизма. При безвозмездном распространении МИ затраты на ее разработку все равно погашаются из государственных источников и создатели как эффективных, так и неэффективных программных средств находятся в близком экономическом положении (иногда создавать плохие изделия даже выгоднее). Обладатель «бесплатно» полученной МИ не может экономически компенсировать свои потери от неработающих изделий; продукты труда программистов и информатиков безвозмездно отчуждаются, не защищаются законом, поэтому никто, кроме энтузиастов, не заинтересован доводить их до товарного вида.

В настоящее время предприняты конкретные шаги по устранению таких печальных итогов политики безвозмездного распространения МИ, как мизерные тиражи программных средств, неразвитость систем автоматизации проектирования, отсутствие экспертных систем и т. п.

Постановлением ГКВТИ СССР от 21 марта 1988 г. № 5 утверждены Типовой договор на создание и использование программных средств для персональных ЭВМ, а также Положение об авторском вознаграждении за издание (тиражирование) и использование программных средств для персональных ЭВМ по авторским договорам. В целях экспериментальной отработки механизма правового и экономического регулирования взаимоотношений разработчиков и потребителей программных средств Московскому экспериментальному вычислительному центру «Элекс» ГКВТИ СССР предоставлено право заключения с разработчиками авторских договоров на создание программных средств ПЭВМ. Тем самым впервые в СССР определенный класс программных средств отнесен к категории объектов охраны авторским правом.

Это лишь первый шаг на пути реализации программы разработки первоочередных нормативных актов в области информатики. В ближайший год должны быть созданы: Типовое (примерное) положение о ВЦ и ВС, Положение о правовом режиме МИ, Положение о порядке разработки, изготовления и поставки МИ, Типовой перечень информационно-вычислительных работ (ИВР) и услуг. Инструкция о порядке установления цен на МИ, Типовой прейскурант на ИВР и услуги, Положение о поставках МИ как продукции, Типовые договоры на поставку МИ, проведение ИВР и оказание услуг и некоторые другие.

Будут разработаны порядок предъявления претензий, связанных с поставкой МИ, проведением ИВР и оказанием услуг, свод нормативных актов, регламентирующих промышленное производство МИ; определены механизмы установления обстоятельств, связанных с возникновением, изменением и прекращением информационных правоотношений, система ответственности за причинение вреда и посягательства на ВС и т. д.

Особенностью создаваемого свода нормативных документов является широкое обсуждение положенных в его основу принципов и вырабатываемых решений — они рассматривались в 1987—1988 гг. на трех всесоюзных совещаниях и международной конференции [10]. В ноябре 1988 г. на Первой Всесоюзной конференции по правовым и социальным проблемам информатики детально обсуждены первые итоги работы, поставлены задачи на 1989 г.

Развернутые работы по созданию законодательства информатики — серьезный шаг по включению экономических факторов, выраженных в юридической форме, в ускорение процесса информатизации в стране. Представ-

* Т. е. не существующего, но позволяющего в рамках принятой условности упростить юридические конструкции.

ляется, что назрел момент для координации усилий всех стран — членов СЭВ в развитии рынка МИ, упорядочении возникающих при этом общественных отношений и выражении их в виде международных правовых актов. Наконец, развивающееся международное сотрудничество с капиталистическими странами требует ускоренного осмысления и применения опыта Всемирной Организации интеллектуальной собственности.

Еще ни одна область знаний не развивалась в темпе, соизмеримом с темпом развития информатики, и вряд ли вина юристов в том, что право информатики стало такой научной проблемой, которая уже в момент зарождения отстала от потребностей практики. Еще никогда в истории производительные силы и вызванные их развитием производственные отношения не изменялись так радикально на протяжении жизни одного поколения.

Право информатики, находящее свое выражение в создаваемом законодательстве, призвано обеспечить экономические интересы людей, стоящих на переднем крае научно-технического прогресса. В обсуждении и развитии поставленных проблем должны участвовать специалисты в области информатики, экономисты, правоведы. Поэтому хотелось бы, чтобы настоящая статья послужила приглашением к сотрудничеству и совместной работе в секции «Нормативно-правовое обеспечение информатики» ГКВТИ СССР.

Именно через экономические интересы быстро формирующейся социальной группы специалистов в области информатики должна найти свое выражение задача развития производства «мягкого товара», именно через интересы информатиков и специалистов, занятых формализацией знаний, можно добиться, ускорения научно-технического и социального прогресса в целом.

103051, Москва, Малый Сухаревский пер., 9, МЭВЦ «Элекс»

1. Карась И. З. Опыт функционирования промышленного предприятия по производству программных средств // Микропроцессорные средства и системы.— 1985.— № 1.— С. 36—41.
2. Блейк Б. Дж. Библиотеки и документальные центры по общественным наукам на североамериканском информационном рынке: В научном цикле международной федерации по документации / М.— ВИНТИ.— 1986.— Т. 11.— № 1.— С. 6.
3. Венгеров А. Б. Право и информатика в условиях автоматизации управления (Теоретические вопросы).— М., 1978.— С. 27—31.
4. Карась И. З. Правовое регулирование общественных отношений в сфере информатики // Сов. гос. и право.— 1987.— № 3.— С. 22—29.
5. Карась И. З. Развитие обязательственных отношений в информатике // Сов. гос. и право.— 1988.— № 4.— С. 92—100.
6. Sartre F. Les caracteristiques "Cle en main" informatique//Mines et micros.— 1985.— N 227.— P. 37—38.
7. Sartre F. L'assurance en informatique: les principes fondamentaux//Mines et micros.— 1985.— N 221.— P. 45—47.
8. Карась И. З. Экономический и правовой режим информационных ресурсов.— Препринт 88-13 ИК АН УССР им. В. М. Глушкова.— Киев.— 1988.— С. 24.
9. Карась И. З. Вопросы правового обеспечения информатики // Микропроцессорные средства и системы.— 1986.— № 1.— С. 3—9.
10. Карась И. З. Информационные ресурсы. Экономические и правовые проблемы. Сб. MSVTI'88 ormane zabezpecenie KPVP.— 1988.— № 1.

Статья поступила 31.10.88

СПРАВОЧНАЯ ИНФОРМАЦИЯ

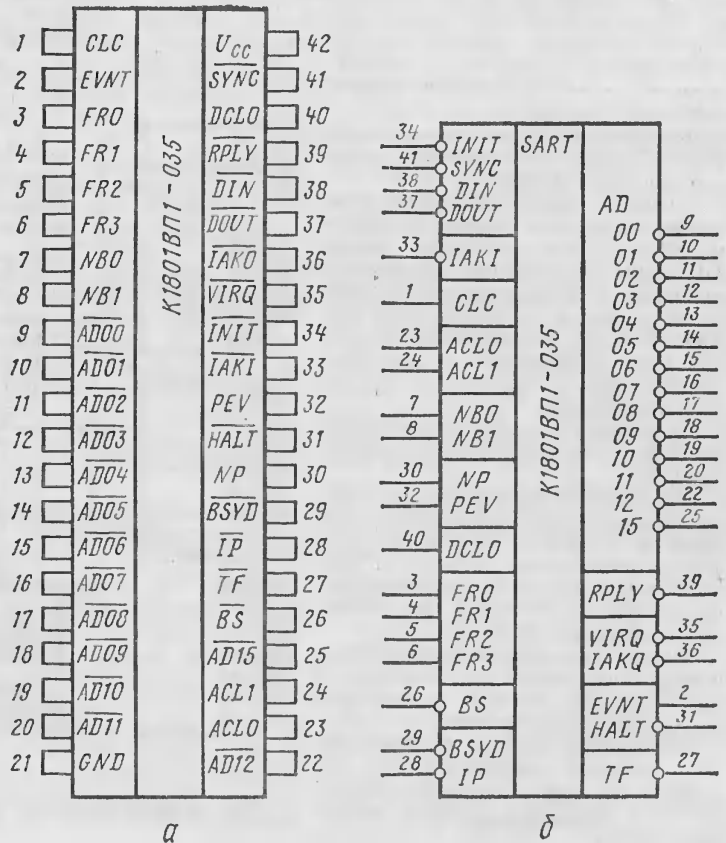
УДК 681.3.01 : 51

ИНТЕРФЕЙСНЫЕ БИС МИКРОПРОЦЕССОРНОГО КОМПЛЕКТА K1801: МИКРОСХЕМА K1801BП1-035

(Продолжение цикла. Начало в № 4—6, 1988)

Микросхема K1801BП1-035 (асинхронный приемник-передатчик), выполненная по p-канальной МОП-технологии, является однокристалльным контроллером внешних устройств, работающих на линии связи с последовательной передачей информации в дуплексном режиме, и предназначена для преобразования параллельной информации в последовательную и наоборот. Кристалл содержит около 5000 транзисторов и помещен в 42-выводной планарный металлокерамический корпус. Условное графиче-

Рис. 1. Условное графическое обозначение микросхемы K1801BП1-035 по порядку расположения (а) и функциональному назначению (б) выводов



Обозначение и наименование выводов

Вывод	Обозначение	Тип	Наименование
1	CLC	Вход	Тактовая частота
2	EVNT	Выход	Прерывания по таймеру 50 Гц
3...6	FR0...FR3	Вход	Выбор скорости обмена
7,8	NB0, NB1	»	Выбор формата
9...20	AD 00...AD 11	Вход-выход, вход	Разряды адреса-данных
21	GND	—	Общий
22	AD12	Вход-выход	Двенадцатый разряд адреса-данных
23	ACL0	Вход	Выбор адреса
24	ACL1	»	
25	AD15	Выход	Пятнадцатый разряд адреса-данных
26	BC	Вход	Выбор внешнего устройства
27	TF	Выход	Сигнал передатчика
28	IP	Вход	Сигнал приемника
29	BSYD	»	Сигнал занятости последовательного канала
30	NP	Вход	Установка паритета
31	HALT	Выход	Останов
32	PEV	Вход	Установка четности или нечетности
33	IAKI	»	Сигнал предоставления прерывания по каналу
34	INIT	»	Установка системной магистрали
35	VIRQ	Выход	Требование векторного прерывания
36	IAKO	»	Предоставление прерывания системной магистрали
37	DOUT	Вход	Вывод данных
38	DIN	»	Ввод данных
39	RPLY	Выход	Ответ
40	DCLO	Вход	Авария источника питания
41	SYNC	»	Обмен
42	U _{CC}	—	Напряжение источника питания

ческое обозначение микросхемы приведено на рис. 1, назначение выводов показано в табл. 1.

В режиме обмена информацией по последовательному каналу микросхема обеспечивает требования интерфейса для радиального подключения устройств с последовательной передачей информации (ИРПС). В составе ЭВМ она может использоваться для связи процессора с удаленными устройствами, в частности с пультовым терминалом.

Микросхема состоит из следующих основных функциональных блоков (рис. 2):

блока системной магистрали, обеспечивающего связь параллельного канала и регистров микросхемы для записи информации (адресов и данных) в регистры, а также чтения информации из регистров или чтения векторов прерываний;

блока синхронизации для записи и чтения информации из регистров и синхронизации работы микросхемы по системной магистрали;

компаратора адресов и управляющих сигналов, предназначенных для выбора по адресу регистров микросхемы и выработки сигналов записи или чтения;

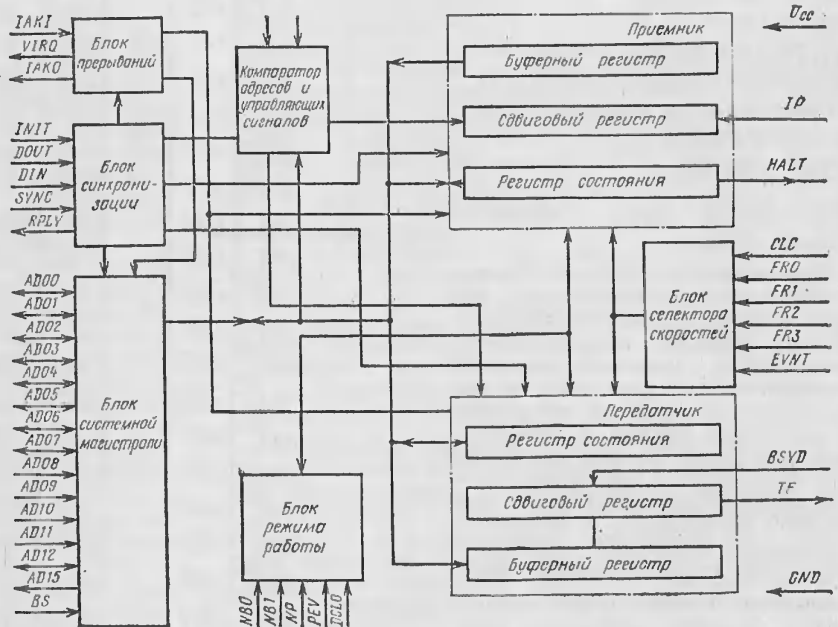


Рис. 2. Электрическая структурная схема устройства последовательного ввода-вывода

Адреса регистров и векторов

Регистры, источник адреса	Группа			
	1	2	3	4
Регистр состояния приемника	177560	176560	176570	XXXXX0
Буферный регистр приемника	177562	176562	176572	XXXXX2
Регистр состояния передатчика	177564	176564	176574	XXXXX4
Буферный регистр передатчика	177566	176566	176576	XXXXX6
Адрес вектора прерывания приемника	060	360	370	XX0
Адрес вектора прерывания передатчика	064	364	374	XX4

блока режимов работы, поддерживающего работу микросхемы по последовательному каналу в различных форматах данных, а также с паритетом или без него;

блоков приемника и передатчика, включающих регистры состояния, сдвиговый, буферный и параллельный;

блока селектора скоростей, обеспечивающего работу микросхемы на разных скоростях обмена информацией по последовательному каналу, а также выработку сигнала EVNT частотой 50 Гц.

Скорости обмена информацией при работе по последовательному каналу с частотой 4608 кГц составляют 50, 75, 100, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 19200 бод. Прием и выдача посылки осуществляются в форматах 5, 7 или 8 информационных битов. Микросхема обеспечивает формирование двухстоповых и прием полустоповых битов (в формате посылки 5 бит), а также формирование и контроль бита паритета (четности или нечетности) и работу без бита паритета.

Микросхема содержит четыре регистра и два источника адресов вектора прерывания. Адреса регистров и векторов — сменные по группам (табл. 2).

Форматы регистров

Регистр состояния приемника (рис. 3). Разряды 0...5, 8...11, 13, 14 не используются и читаются как Лог. 0. Разряд 6 — разрешение работы приемника по прерыванию. Если этот разряд установлен в состоянии Лог. 1, то прерывание разрешено,

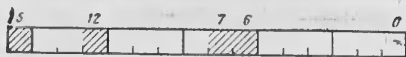


Рис. 3. Формат регистра состояния приемника

если — в состоянии Лог. 0, то запрещено. Сбрасывается в Лог. 0 по сигналу INIT. Доступен по записи и чтению.

Разряд 7 — флаг состояния приемника. Устанавливается в Лог. 1 при поступлении посылки в буферный регистр приемника от предыдущей посылки или от исходного состояния СТОП на линии при включении источника питания после приема первой посылки. Сбрасывается в Лог. 0 по окончании чтения посылки буферного регистра приемника или по сигналу INIT. Доступен по чтению.

Разряд 12 — ошибка переполнения. Устанавливается в Лог. 1, если в сдвиговый регистр приемника поступило более одной посылки без чтения из буферного регистра приемника первой поступившей посылки. При этом независимо от числа поступивших в канал приемника посылок в

буферном регистре сохраняется первая посылка. Разряд 12 находится в состоянии Лог. 0, если поступившая в буферный регистр посылка читается до окончания поступления в сдвиговый регистр последнего информационного бита следующей посылки. Сбрасывается в Лог. 0 по окончании чтения буферного регистра приемника или по сигналу INIT. Доступен по чтению.

Разряд 15 — ошибка в принятой посылке. Устанавливается в состоянии Лог. 1, если есть ошибка паритета в принятой посылке и установлено условие контроля паритета. Находится в состоянии Лог. 0, если нет ошибки паритета или не установлено условие контроля паритета. Сбрасывается в Лог. 0 по чтению буферного регистра приемника или по сигналу INIT. Доступен по чтению.

Буферный регистр приемника (рис. 4)

Разряды 0...7 содержат посылку, принятую с линии. В разряде 0 находится первый бит, в разряде 7 — восьмой бит посылки. При установленном контроле паритета в следующем за последним битом посылки находится бит паритета. При формате 8 бит он анализируется микросхемой, но в буферный регистр не поступает.

Низкий уровень бита в посылке на входе IF соответствует Лог. 1, высо-

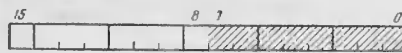


Рис. 4. Формат буферного регистра приемника

кий уровень — Лог. 0 в буферном регистре приемника. Разряды 8...15 не используются и читаются как Лог. 0. Регистр доступен по чтению.

Регистр состояния передатчика (рис. 5)

Разряд 0 — разрыв линии. Если он находится в состоянии Лог. 1, то на выходе TF устанавливается высокий уровень (состояние СТАРТ) при условии, что на входе BSYD — низкий

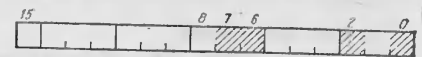


Рис. 5. Формат регистра состояния передатчика

уровень. Если на входе BSYD высокий уровень, то на выходе TF устанавливается низкий уровень (состояние СТОП). Установка разряда 0 в состояние Лог. 1 этот уровень не изменяет. Разряд 0 доступен по записи и чтению.

Разряд 2 — проверка работы. Если этот разряд установлен в состояние Лог. 1, то выдаваемая с выхода TF посылка поступает также и на канал приемника. При этом вход IP для приема посылки с линии закрыт. Разряд 2 доступен по записи и чтению.

Разряд 6 — разрешение работы передатчика по прерыванию. Если этот разряд установлен в состояние Лог. 1, то прерывание разрешено, если в Лог. 0, то запрещено. Доступен по записи и чтению.

Разряд 7 — флаг состояния передатчика. Устанавливается в Лог. 1 по началу выдачи посылки на линию или по сигналу DCLO. Сбрасывается в Лог. 0 по записи данных в буферный регистр передатчика. Если разряд 7 установлен в Лог. 1, то буферный регистр передатчика будет пуст. Если запись данных в буферный регистр произошла во время выдачи посылки, то следующая посылка начнется сразу же по окончании предыдущей. Доступен по чтению.

Разряды 1, 3...5, 8...15 не используются и читаются как Лог. 0.

Буферный регистр передатчика (рис. 6).

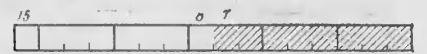


Рис. 6. Формат буферного регистра передатчика

Разряды 0...7 содержат данные для передачи посылки на линию. В разряд 0 записывается первый бит, в разряд 7 — восьмой бит посылки. По окончании записи в этот регистр происходит запись данных в сдвиговый регистр передатчика. При условии низкого уровня на входе BSYD посылка с выхода TF поступает на линию. Бит паритета автоматически следует за последним битом посылки при условии работы с контролем четности или нечетности. Этот бит формируется на выходе TF низким уровнем при записи Лог. 1. и высоким уровнем при записи Лог. 0.

В режиме чтения по адресу буферного регистра передатчика читается адрес источника вектора прерывания. Разряды 8...15 читаются как Лог. 0.

Режим прерывания

Прерывание от приемника. Запрос на прерывание от приемника возникает, если разряды 6 и 7 регистра состояния приемника установлены в 1. При обработке запроса на прерывание читается адрес вектора прерывания приемника.

Прерывание от передатчика. Запрос на прерывание от передатчика возникает, если разряды 6 и 7 регистра состояния передатчика установлены в 1. При обработке запроса на прерывание читается адрес вектора прерывания передатчика.

Формат источников адресов векторов прерываний (рис. 7)



Рис. 7. Формат источников адресов векторов прерываний

Разряд 2 — указатель адреса вектора прерывания. Запись 0 в разряд 2 указывает на адрес вектора прерывания приемника, 1 — передатчика. Разряды 3...7 — сменная часть адреса вектора прерывания. Они читаются как Лог. 0, если выбрана четвертая группа адресов регистров и источников адресов векторов прерываний. Разряды 0, 1, 8...15 не используются, читаются как Лог. 0.

Приоритет запроса от приемника выше приоритета запроса от передатчика.

По окончании приема посылки при отсутствии сигнала СТОП (разрыв линии) на выходе HALT устанавливается низкий уровень. Высокий уровень на этом выходе возникает по сигналу INIT. При тактовой частоте 4608 кГц на выходе EVNT формируются импульсные сигналы со скважностью $Q=2$ и частотой 50 Гц.

Работа микросхемы

Связь микросхемы с процессором осуществляется по каналу МПИ. Временные диаграммы операций

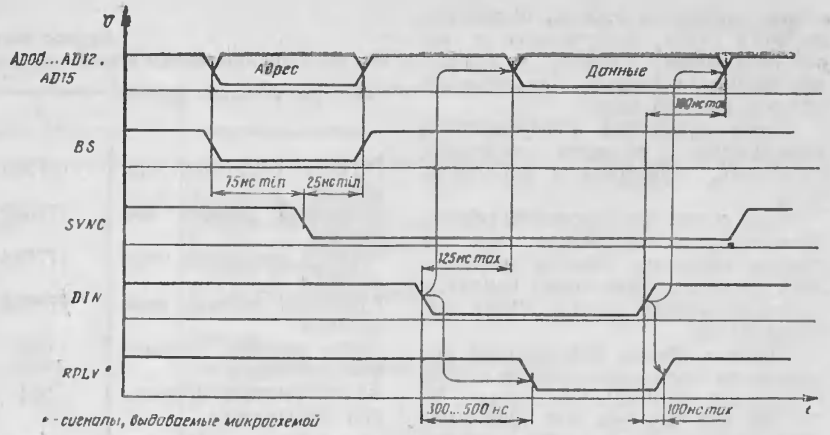


Рис. 8. Временная диаграмма операции чтение (ввод)

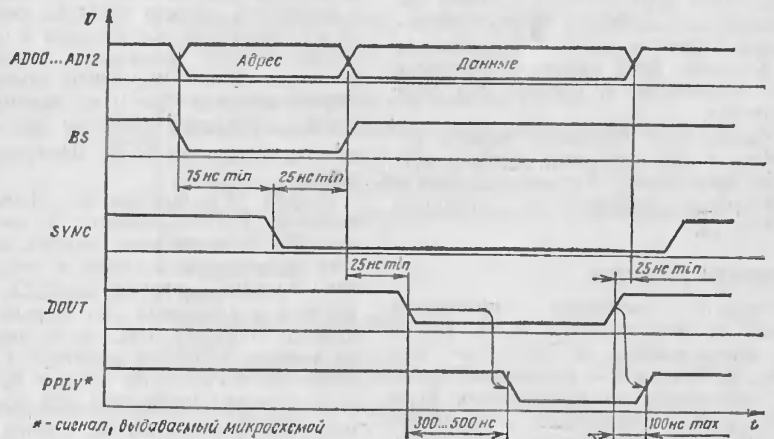


Рис. 9. Временная диаграмма операции запись (вывод)

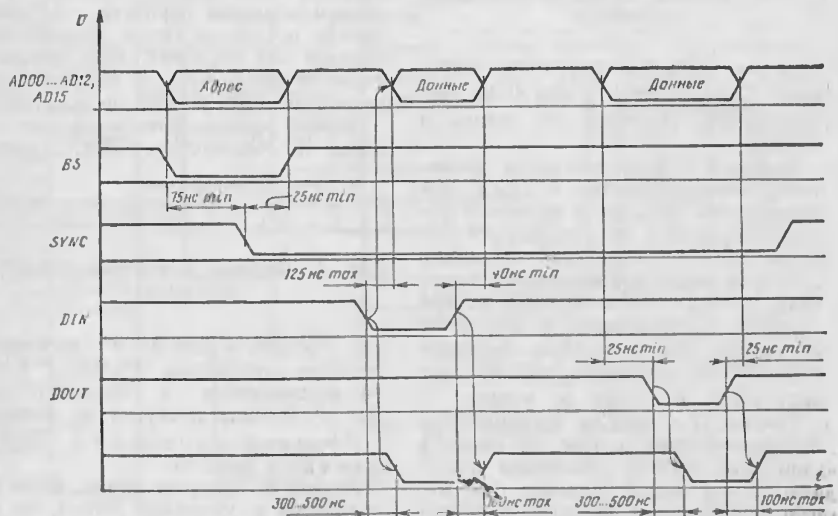


Рис. 10. Временная диаграмма операции чтение-модификация-запись (ввод-пауза-вывод)

тения, записи, чтения-модификации-записи и прерывания приведены на рис. 8—11 соответственно.

Работа приемника

Информация в последовательном коде поступает со входа IP в сдвиговый регистр и по окончании сдвига переписывается в буферный регистр. В разряде 7 регистра состояния приемника выставляется флаг готовности приемника. Если в разряде 6 регистра состояния записано условие работы канала приемника микросхемы по прерыванию, то на выходе VIRQ возникает сигнал запроса на прерывание. Запрос должен быть обработан по системной магистрали процессором, в результате чего по адресу буферного регистра должна быть прочитана посылка. По окончании чтения посылки флаг готовности приемника в разряде 7 сбрасывается и приемник может принимать новую посылку.

При отсутствии условия разрешения прерывания (разряд 6 регистра состояния приемника содержит Лог. 0) требование прерывания на выходе VIRQ не возникает. Процессор должен работать с микросхемой в режиме сканирования (периодического чтения по адресу) регистра состояния приемника и после чтения флага готовности приемника прочитать посылку из буферного регистра приемника.

Пример приема посылки приведен на рис. 12. Чтение флага готовности приемника и ошибки паритета на рис. 12 не показаны. Поступившую в буферный регистр приемника посылку можно читать не позднее поступления в сдвиговый регистр последнего информационного бита следующей посылки, иначе возникает ошибка переполнения (рис. 13). Прием посылок без возникновения ошибки переполнения приведен на рис. 14.

Работа передатчика

В исходном состоянии после подачи низкого уровня на вход DCLO микросхема выставляет в разряде 7 регистра состояния флаг готовности передатчика, свидетельствующий о том, что буферный регистр передатчика пуст. В разряде 6 регистра состояния может быть записано или не записано условие работы передатчика микросхемы по прерыванию. Алгоритм работы передатчика микросхемы по прерыванию или без него аналогичен описанию работы приемника.

Информация из системной магистрали записывается по адресу буферного регистра передатчика. По окончании записи она в параллельном коде переписывается в сдвиговый регистр. При отсутствии сигнала BSYD на выходе TF через $t = 1/16$ длительности бита появляется посылка, автоматически выдвигаемая из сдвиго-

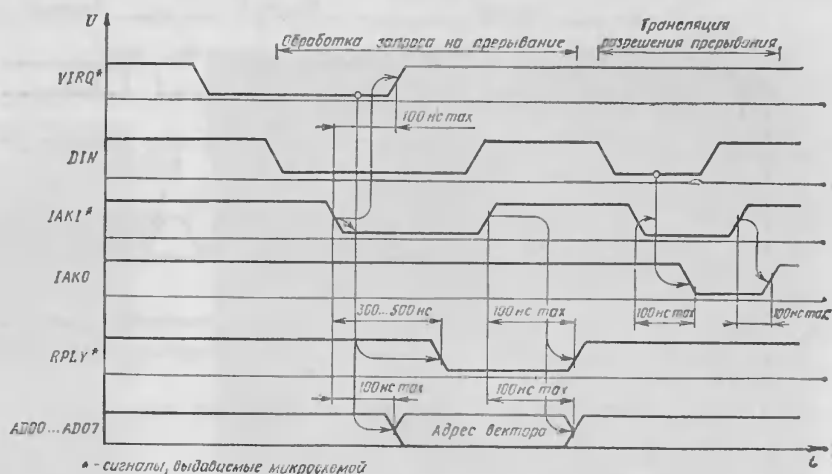


Рис. 11. Временная диаграмма операции прерывания

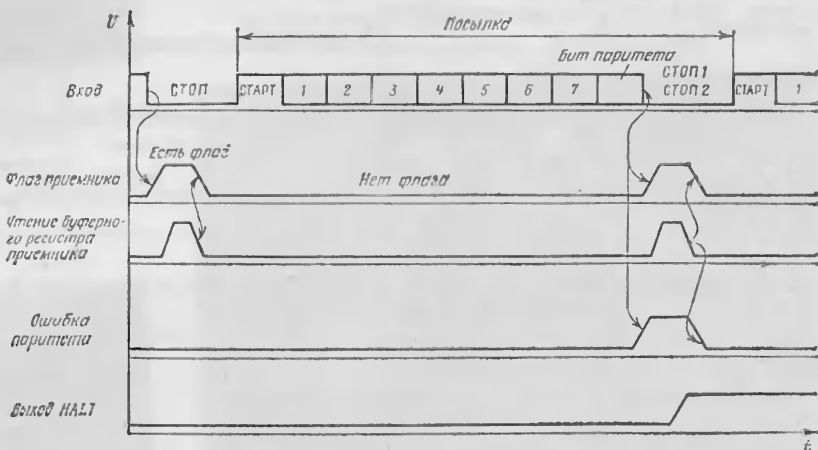


Рис. 12. Временная диаграмма приема седьмой байтовой посылки

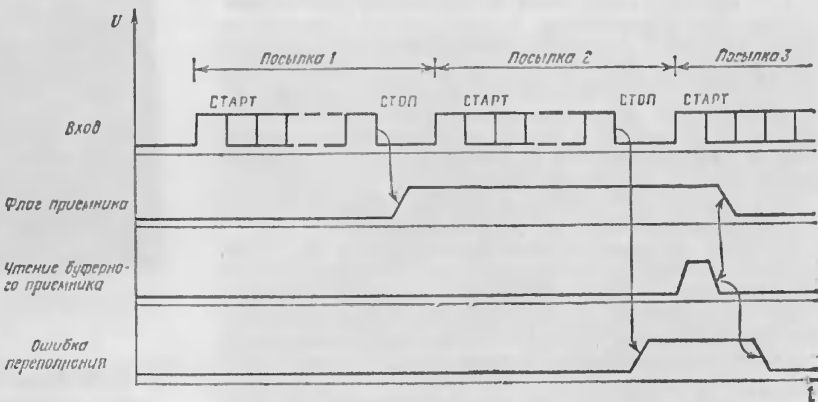


Рис. 13. Временная диаграмма возникновения ошибки переполнения при приеме посылок

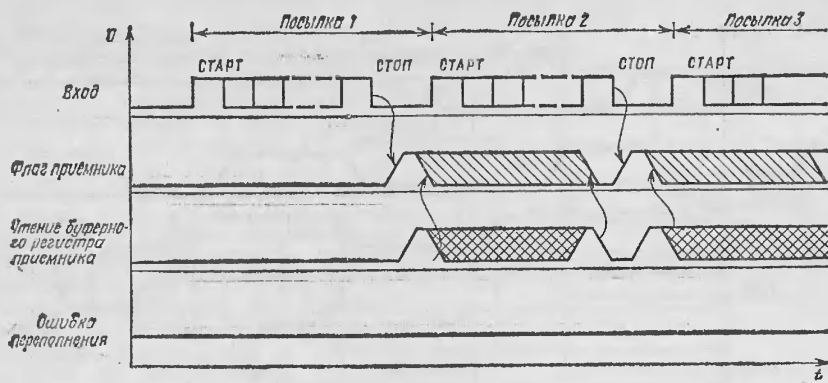


Рис. 14. Прием посылок без возникновения ошибки переполнения

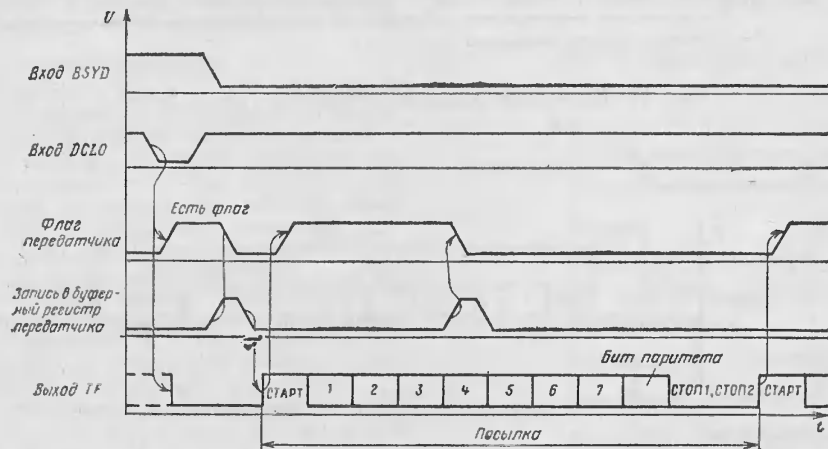


Рис. 15. Временная диаграмма выдачи 7-битовой посылки с формированием бита паритета и двумя СТОП-битами

вого регистра. С началом сдвига посылки в регистре состояния передатчика выставляется флаг готовности, свидетельствующий о том, что буферный регистр передатчика пуст и в него можно записывать новую информацию.

Если запись новой информации произведена во время выдачи посылки на линию, то новая посылка выйдет сразу же после окончания текущей посылки, т. е. после бита СТОП на выходе TF появляется бит СТАРТ новой посылки.

Пример выдачи посылки показан на рис. 15. Сигнал занятости канала BSYD влияет на выход TF и схему синхронизации канала передатчика. Если на входе BSYD низкий уровень, то канал передатчика работает, как описано выше. Если на вход BSYD подать высокий уровень, то на выходе TF устанавливается низкий уровень (СТОП) независимо от того, выходит посылка или нет, и тормозится выдача посылки. Если во время действия высокого уровня на входе BSYD происходит запись данных в буферный регистр передатчика, то эти данные не переписываются в сдвиговый регистр передатчика. После снятия высокого уровня со входа BSYD записанная в буферный регистр посылка переписывается в сдвиговый регистр и на выходе TF через $t = 1/16$ длительности бита появляется бит СТАРТ данной посылки и возникает новый флаг готовности передатчика.

Г. Г. Глушкова

Телефон 208-73-23, Москва

(Окончание. Начало см. на с. 57, 67)

структуризовать информацию произвольным по глубине, ширине, размеру единицы (узла) образом; контролировать доступ к информации, структура которой никоим образом не связана со структурой коллектива людей, обращающихся к ней (стандартные средства контроля доступа хороши как дополнение к ДФС для защиты самой ОС и (или) файловых систем небольших размеров, используемых небольшими коллективами).

ДФС работает около двух лет на шести ЭВМ «Электроника 79», объединенных в сеть, с 13—14 терминалами на каждой и более 100 постоянными пользователями. Средствами ДФС на каждой пятой инструментальной ЭВМ создана разветвленная файловая система, используемая для коллективной разработки ПО. На одной из ЭВМ создан и поддерживается архив системного и прикладного ПО в основном для ОС РАФОС общим объемом около 300 Мбайт.

Дистрибутив ДФС готов к передаче другим пользователям и уже передан в две организации.

Средства организации файловых систем в ОС Unix, MS/DOS, ставших стандартными для компьютеров соответствующих классов, обладают теми же характерными особенностями (естественно, по другому реализованными): богатыми возможностями структуризации информации, динамическим распределением памяти, средствами защиты доступа.

141570, Моск. обл., Солнечногорский р-н, пос. Менделеево, ул. Институтская, 18, кв. 47, Тимофеев Евгений Сергеевич; тел. 532-83-71

(Окончание. Начало см. на с. 85)

ЛИТЕРАТУРА

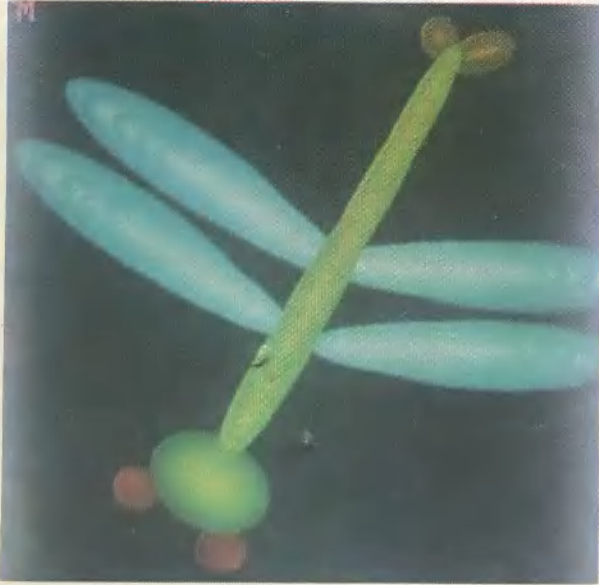
1. Ромм М. Беседы о кино.— М.: Искусство, 1964.
2. Кочетков Г. Б. Компьютерные игры: свет и тени// Микропроцессорные средства и системы.— 1985.— № 3.
3. Pournelle J. Expanded/Extended Memory.— Byte.— 1986.— № 5.
4. Родионов А. Б. Персональный компьютер в музыкальном творчестве// Микропроцессорные средства и системы.— 1987.— № 3.
5. Puscce G. A. Elements of object oriented programming.— Byte.— 1986.— № 8.
6. Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации// Микропроцессорные средства и системы.— 1986.— № 1.

Статья поступила 24.03.87.

ПОПРАВКА

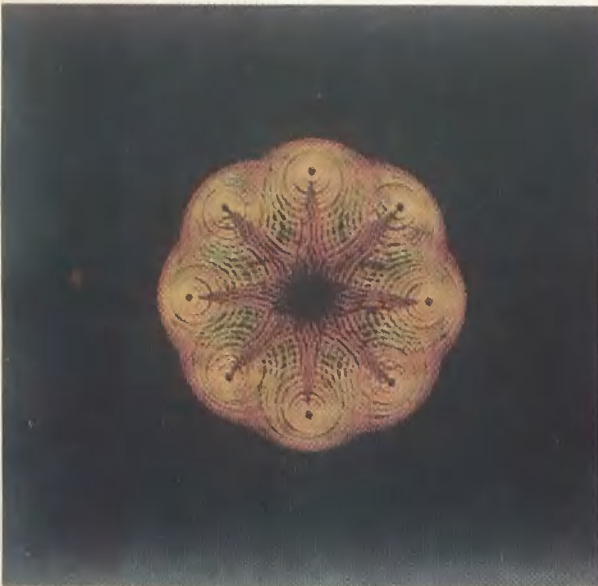
В № 4 за 1988 г. на второй полосе вкладки в материале Кушнира В. Е. на месте нижнего слайда должен быть слайд Злачевского А. Е. из № 5 за 1988 г. с первой полосы вкладки и наоборот.

(Окончание. Начало см. на 2 с. обложки)



Построение изображений копированием с преобразованием псевдообъемных примитивов и фигур

Геометрическим и цветовым преобразованиям могут подвергаться псевдообъемные примитивы и фигуры, нарисованные с помощью плоских и псевдообъемных кистей. Данное изображение построено копированием эллипсоидов с цветовыми и геометрическими преобразованиями.



Построение изображений из сегментов

Примитивы и фигуры, нарисованные кистями, объединяются в именованные сегменты, которые могут подвергаться геометрическим преобразованиям как целое. Данное изображение получено 8-кратным копированием сегмента, построенного, в свою очередь, копированием дуги с выполнением цветовых и геометрических преобразований.



Копирование с преобразованием сегментов и примитивов

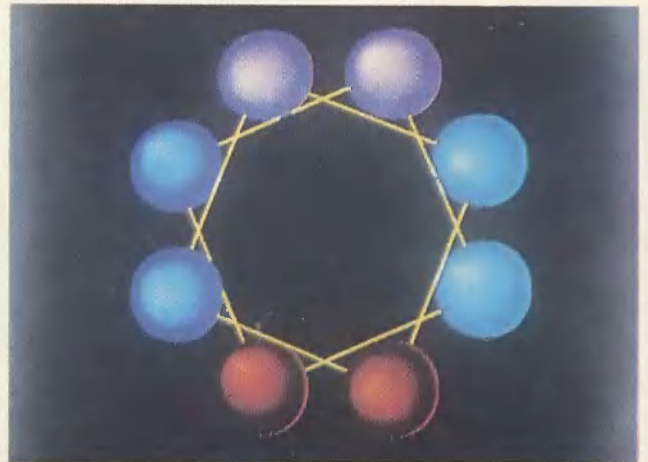


Рисунок построен копированием сегмента, составленного из четырех шаров и ломаной

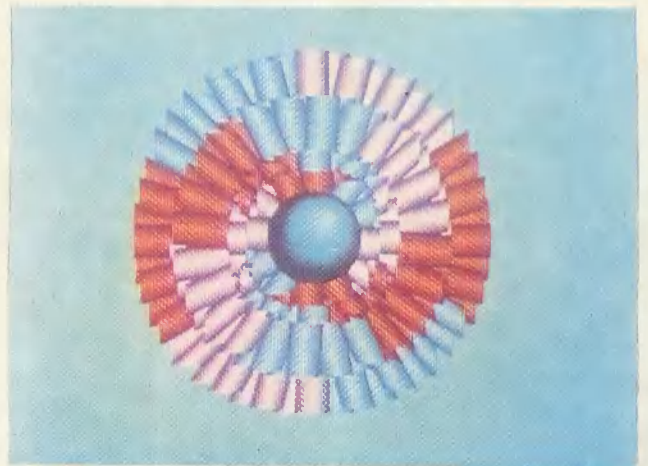


Рисунок построен копированием сегмента, нарисованного объемной кистью

КОМПАН

компьютерные системы

Архитектура «КОМПАН АТ-турбо плюс» базируется на центральном процессоре HARRIS 80286 с тактовой частотой 16/21 МГц и на других лучших технических решениях этой прогрессивной архитектурной линии. Аппаратная реализация на кросс-плате всей оперативной памяти, EGA и VGA адаптеров, параллельных и последовательных каналов ввода-вывода, контроллеров дисков и т. д. позволяет значительно повысить надежность и быстродействие компьютера. Высокая производительность, функциональное разнообразие и емкость памяти (до 8 Мбайт оперативной памяти, 2 устройства на гибких магнитных дисках, жесткие диски до 72 Мбайт), «дружественный интерфейс с пользователем на базе цветной графики и манипулятора типа «Мышь», возможность укомплектования богатым набором печатающих устройств, доступных на западном рынке, модульность и возможность расширения функций делают «КОМПАН АТ-турбо плюс» идеальным техническим средством реализации Ваших сложных производственных, финансовых и научных задач как в режиме автономного использования, так и в структуре локальных сетей. Полная программная и аппаратная совместимость с зарубежными аналогами позволяет использовать громадное по объему и богатству программное обеспечение для этого типа компьютеров. Низкая потребляемая мощность (200 Вт), отсутствие специальных требований к помещениям, гарантийное в течение 24 месяцев и затем абонементное фирменное обслуживание делают «КОМПАН АТ-турбо плюс» доступным для широкого круга организаций и пользователей. Применение BIOS собственной разработки, русского и американского стандартов в клавиатуре, матричных и лазерных принтеров, русифицированных на аппаратном уровне, отечественных и русифицированных программных средств, широкого набора отечественных художественных шрифтов для печати дают «КОМПАН АТ-турбо плюс» неоспоримые преимущества на рынке.

Вас что-то не удовлетворяет в IBM PC? К Вашим услугам наш «КОМПАН АТ-турбо плюс». Оригинальная архитектура, придающая феноменальные скоростные способности, размещенные на одной плате 8 Мбайт оперативной памяти, супер EGA/VGA адаптера, буфер дисковой памяти и собственный КОМПАН-BIOS, реализующий двуязычный режим работы компьютера, — все это выводит «КОМПАН АТ-турбо плюс» в лидеры класса 286. Полная программная и аппаратная совместимость.

«КОМПАН АТ-турбо плюс»

СОСТАВ УСТРОЙСТВ

- ПРОЦЕССОРНЫЙ БЛОК
- МОНИТОР
- КЛАВИАТУРА
- МАНИПУЛЯТОР ТИПА «МЫШЬ»
- ПРИНТЕР

ПРОЦЕССОРНЫЙ БЛОК

Кросс-плата

- 16-разрядный микропроцессор HARRIS 80286 на 16/21 МГц
- 7 каналов прямого доступа в память
- 15 уровней прерывания
- Сопроцессор INTEL 80287 на 6, 8, 10 или 16 МГц
- от 512 Кб до 8 Мб ОЗУ
- Часы-календарь с встроенным (3,6 В) или внешним (6 В) батарейным питанием
- Дешифраторы контроллеров для 2-х гибких и 2-х жестких дисков
- 2 последовательных (RS-232) и параллельный (Centronix) порты
- EGA и VGA режимы (640×480 точек)

Дополнительные платы

- Адаптер дисплея высокого разрешения 1024×1024 точек
- Цифро-аналоговые и аналого-цифровые преобразователи
- Дополнительные RS-232 и CENTRONIX
- Другие совместимые с IBM аппаратные интерфейсы

Устройства памяти

- «Винчестер» 21 Мб, 42 Мб или 72 Мб (время доступа от 28 до 65 мс)
- НГМД-360 Кб/720 Кб/1.2 Мб/1.44 Мб на 3.5" и 5.25"
- Стример встраиваемый 5 Мб/мин, 60 Мб или внешний 5.25 Мб/мин 125 Мб

Источник эл/питания

- Встроенный, мощность 200 Вт, напряжение 220 В (+/-10%), 50/60 Гц
- Непрерываемый (с аккумулятором) на 1,2 или 4 компьютера

МОНИТОР

- EGA или VGA монитор 14"; в графическом режиме 640×480 точек, 16 цветов из палитры 64 оттенка; в текстовом режиме 25 строк по 80 символов
- RGB-монитор 19" на 1040×780 точек
- В/В мониторы с различными характеристиками

КЛАВИАТУРА

- 102 клавиши, в том числе 12 функциональных, двуязычная

МАНИПУЛЯТОР ТИПА «МЫШЬ»

- Электронная или эл/механическая к RS-232 с программным обеспечением

ПРИНТЕР

- Матричные типы FX-850 FX-1050 или LQ 2550
- Лазерный типа HP Laser Jet II

ЕСЛИ НАША ИНФОРМАЦИЯ ВАС ЗАИНТЕРЕСОВАЛА, ПРОСИМ ОБРАЩАТЬСЯ ПО АДРЕСУ:

СОВМЕСТНОЕ СОВЕТСКО-ЗАПАДНОГЕРМАНСКОЕ ПРЕДПРИЯТИЕ

СССР, Ленинград, 198092, Маршала Говорова, 52. тел. 2521773, 2521564, телекс 121412, телефакс 2524184, а/я 84—260, р.сч. 608019038 в ЛОУ ВЭБ

1
МОСКВА • 1989
МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ