МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

2 / 1989

ISSN 0233-4844

Универсальный лабораторный комплекс для вузов страны — учебная гибкая производственная система

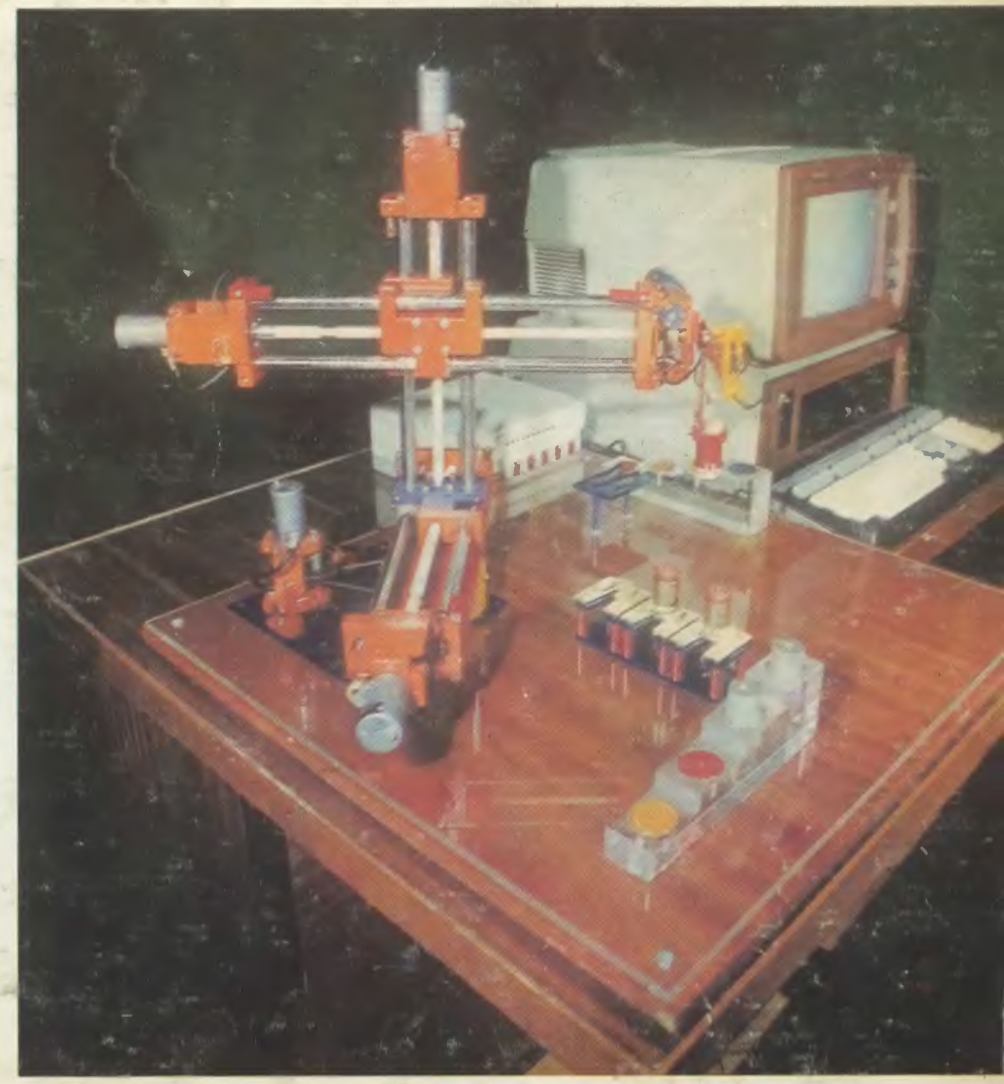
Комплект БИС серии КМ1823 для построения МП-систем управления двигателями внутреннего сгорания с искровым зажиганием

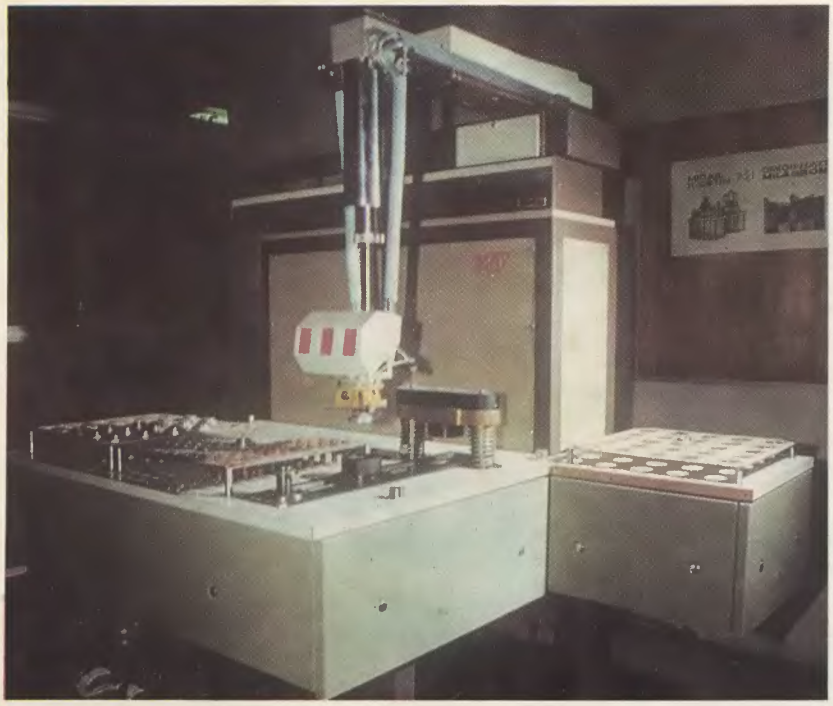
Комплекс для анализа и обработки изображений на базе мультипортовой памяти и спецпроцессов

СУБД RDBMS — удобное и простое средство управления данными на микроЭВМ

Компьютерные вирусы. Средства защиты от них и сервисная служба диагностики и лечения

Объектно - ориентированное программирование — новая альтернатива технологии традиционного программирования





Учебная ГПС

УЧЕБНАЯ ГПС

(к ст. Макарова И. М. и др.)

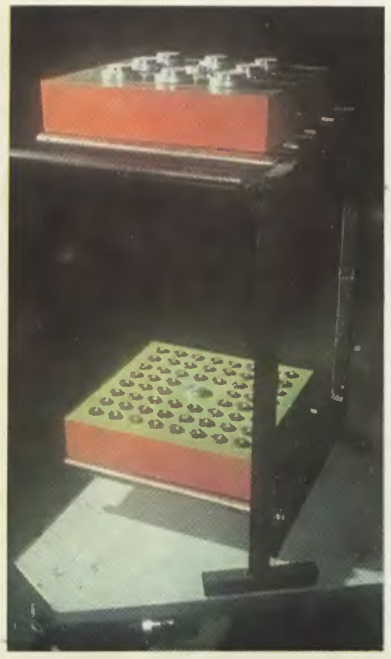
Учебная ГПС — универсальный лабораторный комплекс, разработанный на кафедре «Проблемы управления» МИРЭА для изучения основных дисциплин специальности «Робототехнические системы и комплексы» в вузах, техникумах и предприятиях, осуществляющих подготовку и переподготовку кадров по робототехнике. В ее состав входят: автоматизированный учебный склад; транспортная тележка; центр механообработки на базе токарного и фрезерного станков с ЧПУ, загружаемых роботом МП-9С деталями из накопителя; три сборочных центра:

специализированный, предназначенный для сборки и запрессовки изделий типа «вал-втулка» (робот «Электроника НЦТМ-01М» и пресс);

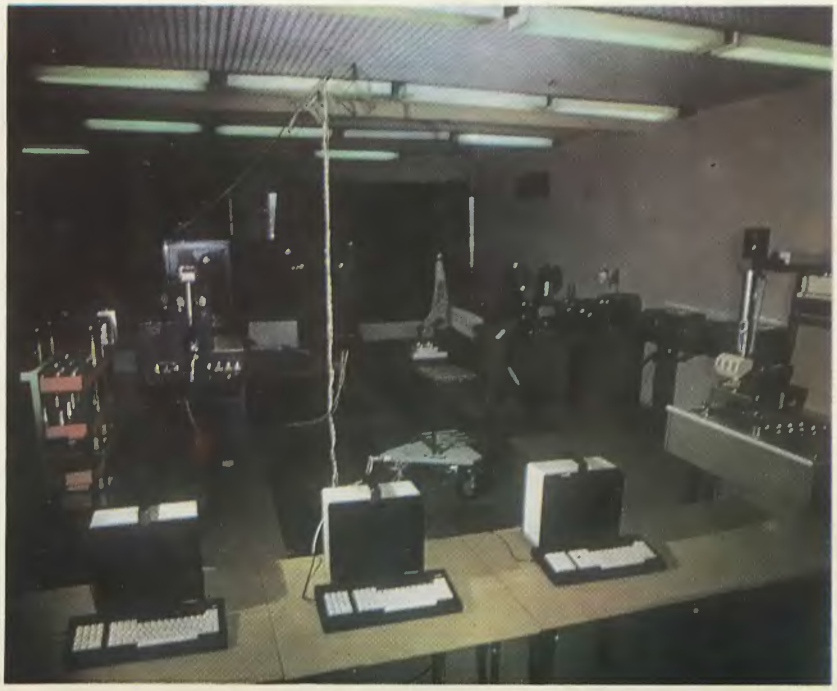
универсальный (робот «Электроника НЦТМ-01» с электрическим захватным устройством, поворотный стол и накопитель деталей);

адаптивный (два робота РМ-01 и система технического зрения).

(Продолжение на 3 с. обл.)



Транспортная тележка



Специализированный сборочный центр

Пролетарии всех стран, соединяйтесь!

ОРГАН ГОСУДАРСТВЕННОГО КОМИТЕТА СССР ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ И ИНФОРМАТИКЕ

Издается с 1984 года

МП МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 2/1989 МОСКВА

СОДЕРЖАНИЕ

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Шмат В. К., Усов Н. Н.— БИС форматора данных для управления СБИС ЗУ ЦМД . . . 2
Шмат В. К., Кокорев Д. В., Усов Н. Н.— Контроллеры внешних запоминающих устройств на ЦМД . . . 6
Лангуев В. В., Болотов С. А., Ольшак А. И.— Комплект БИС серии КМ1823 для микропроцессорных систем управления двигателями внутреннего сгорания . . . 12

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Крюков В. А.— Анализ принципов объектно-ориентированного программирования . . . 14
Абрамов С. М., Пименов С. П., Абакумов А. А., Хаткевич М. И.— Компьютерный вирус . . . 22

Базы данных

Акулов В. К., Журавлев В. И., Новожилов Н. А., Шинкевич С. Л.— Система управления реляционными базами данных . . . 24
Волков Д. К.— МИКРОСУБД на языке БЕЙСИК для ПЭВМ . . . 26
Максимяк С. П., Сорокин Ю. Ю., Субач В. В.— Непосредственный доступ к данным в файлах на Паскале-1 . . . 28

САПР цифровых устройств на ПЛИС

Красильников И. В., Приходько П. С., Чурков В. М., Щетинин Ю. И.—Микросхемы программируемой матричной логики серии КР1556 . . . 31
Чурков В. М., Красильников И. В.— Подсистема проектирования на ПМЛ . . . 34
Чурков В. М., Котрелев С. А., Захарова Л. Е.— Программное обеспечение САПР цифровых устройств для программируемой матричной логики серии КР1556 . . 35
Чурков В. М., Захарова Л. Е.— Подсистема проектирования ПЛМ . . . 41
Чурков В. М., Котрелев С. А.— Подсистема проектирования на ЭПЛК . . . 42

Машинная графика в САПР

Вельтмандер П. В., Жуков Г. В., Килина Л. В. и др.— Системы автоматизированного конструирования для персонального АРМ . . . 46
Голиков К. П., Педанов И. Е., Щурий В. Н.— Подготовка инженерной документации с помощью системы РЕДГРАФ . . . 49
Бобков В. А., Кислюк О. С., Боровина Т. А.— Диалоговая графическая система для исследования структур сложных молекулярных объектов МОЛЛЮСК-2 . . 52

Форум «МП»

Иванов С. М., Шахнов В. А.— Достижения НРБ в области микропроцессорной техники и программного обеспечения . . . 56
Лишак Е. В.— Байтоторговля: прошлое, настоящее, будущее . . . 60

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Робототехнические комплексы

Сайкин В. Т., Артюх И. Г., Карпов И. И., Юданов В. А.— Учебный робот . . . 68
Макаров И. М. и др.— Учебная гибкая производственная система . . . 71
Кузьмичев С. В., Набутовский А. М., Селезнев В. П.— Система подготовки программ для программируемых контроллеров . . . 75
Руцков М. В.— Мультипортовая память для построения видеокомплексов . . . 77
Матвеенко В. И., Староверов Ю. Г.— Бинарная система технического зрения для промышленного применения . . . 81
Коврига Д. С., Гладченко С. М.— Расширение программного обеспечения для промышленного робота «Электроника НЦТМ-01» . . . 84
Горбачев В. С., Липпинг В. Э.— Система трехмерной машинной графики для моделирования промышленных роботов . . . 86
Гадалов А. Ф., Борзов Г. В., Ильин Е. П.— Система редактирования, передачи управляющих программ и контроля для станков с УЧПУ «Электроника НЦ-31» . . 88
Водовозов В. М., Горячев Ю. А., Пожидаев А. К.— Комплектная система ЧПУ на базе ДВК . . . 90

Справочная информация

Рекомендации по применению микросхем серии К1801 . . . 91

© «Московский экспериментальный вычислительной центр «Элекс»

УДК 681.323

В. К. Шмат, Н. Н. Усов

БИС ФОРМАТОРА ДАННЫХ ДЛЯ УПРАВЛЕНИЯ СБИС ЗУ ЦМД

Минимальный набор БИС управления, достаточный для построения систем памяти на СБИС ЗУ ЦМД, включает аппаратно зависимый контроллер*, аппаратно независимый форматор данных (ФД) и дополнительно блок защиты информации (БЗИ) (рис. 1).

Контроллер обеспечивает командный интерфейс с ведущей системой, т. е. принимает, дешифрирует и выполняет определенную систему команд (операций) поиска, записи и чтения нужной информации; формирует необходимый набор управляющих сигналов для СБИС ЗУ ЦМД: формирователей поля продвижения доменов (ВМП), репликации и генерации доменов, обмена доменами, фазы обработки данных (строб усилителей считывания), разделения каналов чтения, управления ПЗУ, карт годности (браков). Кроме этого, контроллер вырабатывает сигналы внутреннего интерфейса для синхронизации и управления работой других функциональных блоков, например ФД, обеспечивает страничную адресацию в ЗУ ЦМД и алгоритмы поиска и преобразования виртуальных адресов для представления ЗУ ЦМД пользователя как ЗУПВ с блочной структурой и виртуальными адресами 0, 1, 2..., m — 1, где m — число доменных позиций в регистрах хранения ЗУ ЦМД.

Структура контроллера аппаратно зависит от конкретного типа СБИС ЗУ ЦМД, в то же время идеология построения архитектуры контроллера должна поддерживать общую архитектуру устройства управления, не зависимую от конкретной архитектуры СБИС ЗУ ЦМД.

Для аппаратной независимости от архитектуры ЗУ ЦМД форматор данных должен реализовать общий принцип (алгоритм) распределенной обработки информационных массивов записи-чтения при обмене с памятью на ЦМД. Гибкость архитектуры обеспечивается возможностью его работы с произвольным числом параллельно работающих микросхем памяти. Внутренний интерфейс управляет работой ФД, учитывает конкретную архитектуру ЗУ ЦМД, обеспечивает аппаратную независимость. Его образуют следующие функциональные сигналы: RWR — тип операции, выполняемой СБИС ЗУ ЦМД; $\overline{FL}$ — сигнал об окончании страницы данных; $\overline{SH1}$, $\overline{SH2}$ — сигналы стробов, задающих конкретный тип СБИС ЗУ ЦМД.

Для нормального функционирования в составе устройства управления ФД содержит два интерфейса: интерфейс пользователя 7 для обмена в параллельном коде, например байтами; интерфейс ЗУ ЦМД (шины 8 и 9). По шине 8 ФД имеет доступ к карте годности, а по двунаправленной шине 9 принимает или передает данные к ЗУ ЦМД. В соответствии с текущей информацией о карте годности ФД преобразует информацию для (от) пользователя.

Структурная схема устройства управления, содержащая минимальный базовый набор БИС приведена на рис. 2.

Для преемственности базовой архитектуры БИС контроллеров К1806ВП1-103, 157 поддерживает единую концепцию страничной адресации и поиска информации, не зависящую от типа микросхем памяти. Обмен информацией происходит блоками конечной длины (страницами), образуемыми однотипными позициями доменов в регистрах хранения, поэтому для минимизации времени при обменах приходится отказываться от линейной структуры адресов, т. е. вводить понятие виртуального адреса страницы. Принципы страничной адресации и поиска нужной информации являются решающими при проектировании систем памяти на

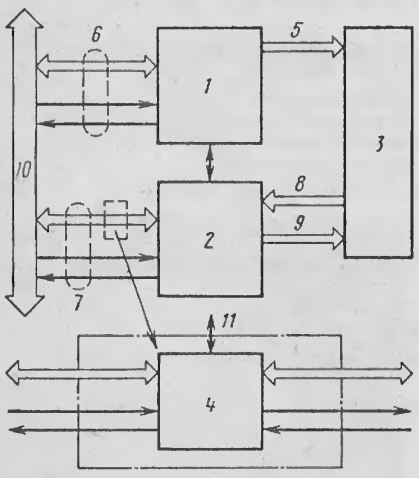


Рис. 1. Базовая структура многофункционального устройства управления ЗУ ЦМД:
1 — контроллер временной последовательности, алгоритмов страничной адресации и поиска, записи-чтения; 2 — форматор данных; 3 — модуль памяти (СБИС ЗУ ЦМД от 1 до n); 4 — блок защиты информации; 5 — интерфейс управления СБИС ЗУ ЦМД; 6 — интерфейс с ведущей системой; 7 — интерфейс обмена данными с системой 10; 8 — шина браков; 9 — шина данных СБИС ЗУ ЦМД; 10 — ведущая система; 11 — внутренний интерфейс устройства управления

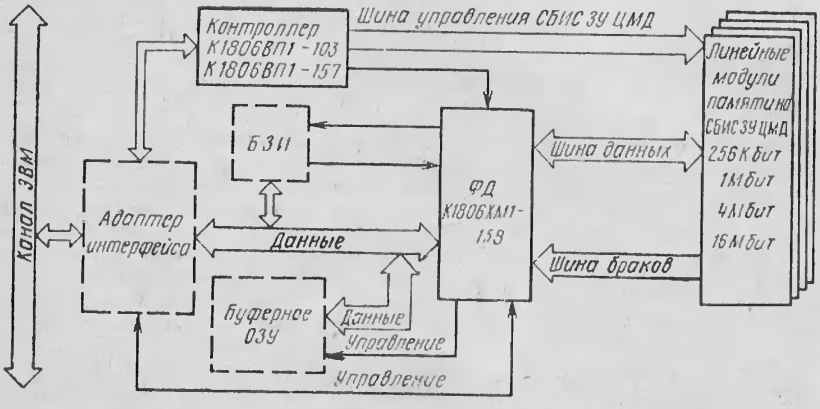


Рис. 2. Структурная схема управления памятью на СБИС ЗУ ЦМД базового комплекта

* Шмат В. К. Специализированные БИС управления для СБИС ЗУ ЦМД. // Микропроцессорные средства и системы.— 1987.— № 6.— С. 3—9.

ЦМД, так как они не только задают временные параметры системы, но и определяют аппаратные и программные средства реализации алгоритмов адресации и поиска.

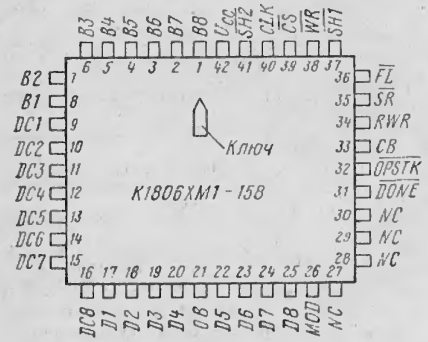
БИС форматора данных К1806ХМ1-158

Микросхема ФД обеспечивает формирование массивов данных при выполнении информационных операций обмена от одной до восьми параллельно-работающих СБИС ЗУ ЦМД различной архитектуры с учетом данных о бракованных регистрах этих СБИС (карта годности). Условное графическое обозначение микросхемы приведено на рис. 3, назначение выводов БИС показано в таблице, структурная схема дана на рис. 4.

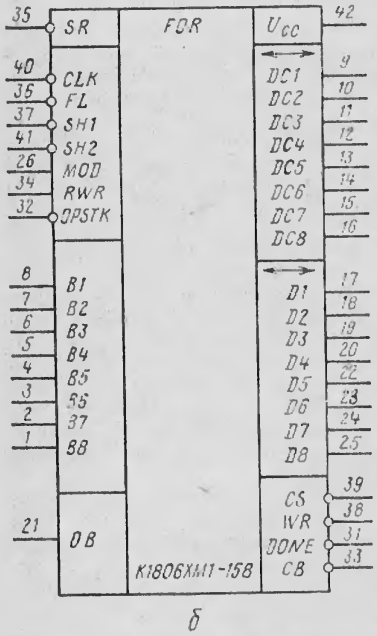
Высокий уровень входных и выходных сигналов соответствует значению

Назначение выводов микросхемы К1806ХМ1-158

Вывод	Обозначение	Назначение
1...8	B8...B1	Выходы разрядов браков
9...16	C1...C8	Входы-выходы разрядов данных ЦМД
17...20, 22...25	DC1...DC8	Входы-выходы разрядов данных системы
21	0B	Общий вывод
26	MOD	Вход «ОЗУ — стек»
27...30	NC	Не используются; подключить к шине 0В
31	DONE	Выход «стек готов»
32	OPSTK	Вход «операция стека»
33	CB	Вход «управление шиной»
34	RWR	Вход «запись-чтение»
35	SR	Вход «начальная установка»
36	FL	Вход «завершено»
37	SH1	Вход «строб-1»
38	WR	Выход «запись ОЗУ»
39	CS	Выход «разрешение ОЗУ»
40	CLK	Вход внешнего тактового генератора
41	SH2	Вход сигнала «строб-2»
42	U_{cc}	Напряжение источника питания 5 В



а



б

Рис. 3. Условное графическое изображение микросхемы К1806ХМ1-158: по порядку расположения (*а*) (NC — не используемые выводы) и функциональному назначению (*б*) выводов.

Лог. 1. Сигнал SR=0 обеспечивает начальную установку БИС в детерминированное состояние и сброс указателей адресов СУС. Сигнал FL=0 вырабатывается ведущей схемой базового комплекта БИС К1806ВП1-103, 157 по окончании информационной операции с каждой страницей ЗУ ЦМД и действует аналогично сигналу SR.

Сигнал RWR указывает тип информационной операции с ЗУ ЦМД, при операции ЗАПИСЬ RWR=0. По сигналу SH1=0 информация о текущих регистрах хранения ЗУ ЦМД (карта годности), поступающая по шине В через буфер БФ1, заносится в регистр РГ1. В режиме ЧТЕНИЕ происходит также занесение данных, поступающих из ЗУ ЦМД по шине DC через буфер БФ2 в регистр РГ2. В течение «временного окна» (SH2=0) происходит побитная обработка данных внутри форматора, которая заключается в последовательном обмене данными между регистрами РГ2 и РГ3 с учетом содержимого регистра РГ1. При этом в

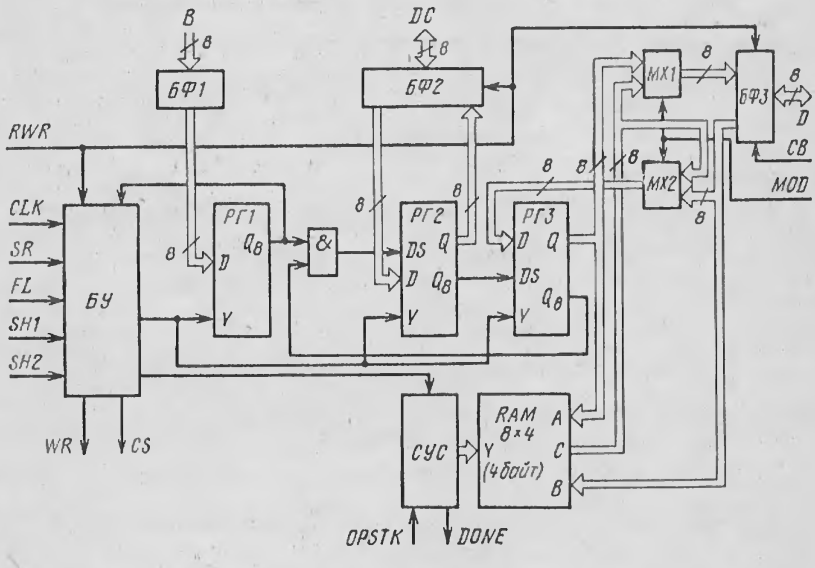


Рис. 4. Структурная схема микросхемы К1806ХМ1-158:
БУ — блок управления сдвигами и формирования байта и схема синхронизации; БФ1, БФ2 — входной и выходной буферы шины браков ЗУ ЦМД; РГ1, РГ2, РГ3 — сдвиговые регистры браков, данных ЗУ ЦМД и данных системы соответственно; СУС — схема управления внутренней памятью БИС; RAM — двухпортовое СОЗУ емкостью 4 байт (организация типа FIFO); MX1, MX2 — мультиплексоры выходных и входных данных системной шины; БФ3 — двунаправленный тристабильный буфер байтовой системной шины

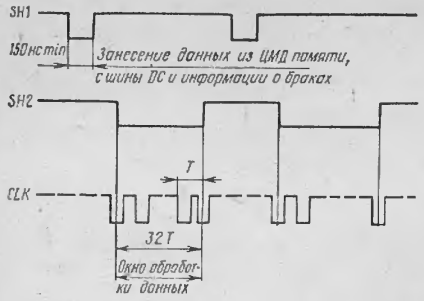


Рис. 5. Временная диаграмма формирования сигналов SH1 и SH2.

режиме ЧТЕНИЕ негодные позиции бракованных регистров отбрасываются и происходит уплотнение информации, а в режиме ЗАПИСЬ соответствующие позиции заполняются нулями, т. е. происходит распределение информации. При этом значению 0 на шине DC соответствует запрет записи домена (генерации) в ЗУ ЦМД.

Сигналы SH1 и SH2 вырабатываются ведущей схемой базового комплекта. При использовании ФД вне базового комплекта необходимо обеспечить их формирование в соответствии с определенной временной диаграммой (рис. 5).

Занесение и обработка данных происходят под управлением БУ. По окончании обработки каждого байта (формирования байта при ЧТЕНИИ или вывода байта на шину DC при ЗАПИСИ) БУ вырабатывает импульсный запрос передачи CS=0, длительность которого равна двум периодам внешнего тактового генератора CLK, и флаг готовности к передаче DONE=0. В режиме ЧТЕНИЕ дополнительно вырабатывается импульсный сигнал WR=0, длительность которого равна периоду тактового генератора CLK.

Сигнал СВ используется для дополнительного управления шиной D. Режим ее работы задается типом операции RWR: ЧТЕНИЕ-передача, ЗАПИСЬ-прием. При СВ=1 шина D всегда работает на прием (состояние «отключено»).

Вход MOD посредством переключения мультиплексоров MX1 и MX2 задает режим работы системной шины непосредственно на регистр РГ3 или через внутреннее СОЗУ, выполненное как магазин с процедурой FIFO и двумя независимыми каналами записи-считывания.

Если MOD=1, то пользователь обязан предоставить (получить) один байт данных по шине D в течение вырабатываемого микросхемой БУ им-

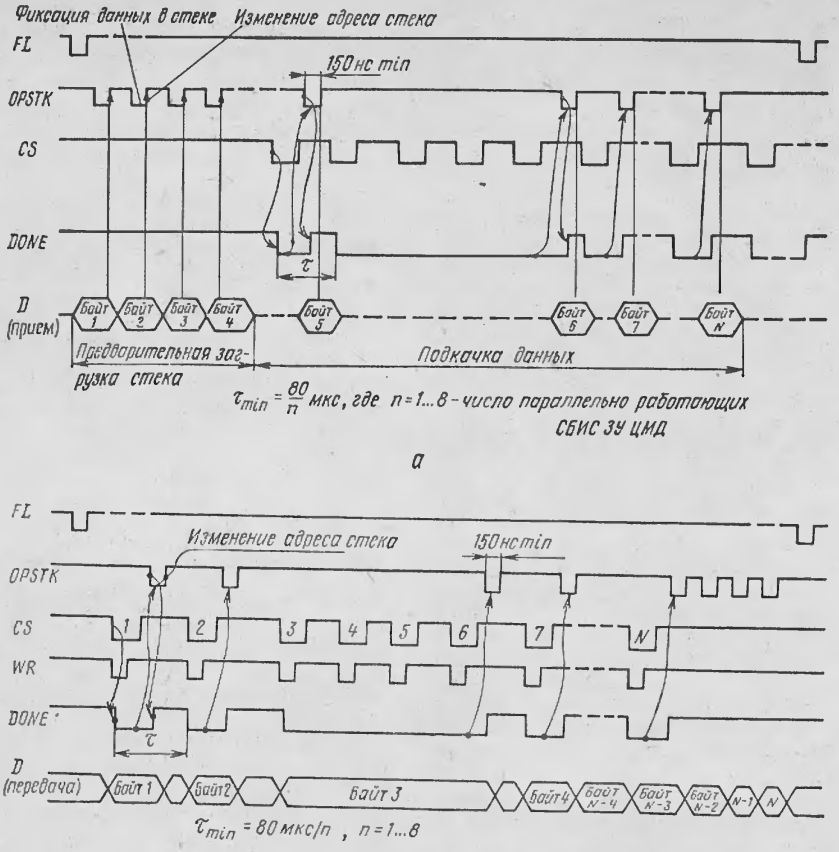


Рис. 7. Временные диаграммы работы микросхемы через стек (МОД-0) в режимах записи (*а*) и чтения (*б*)

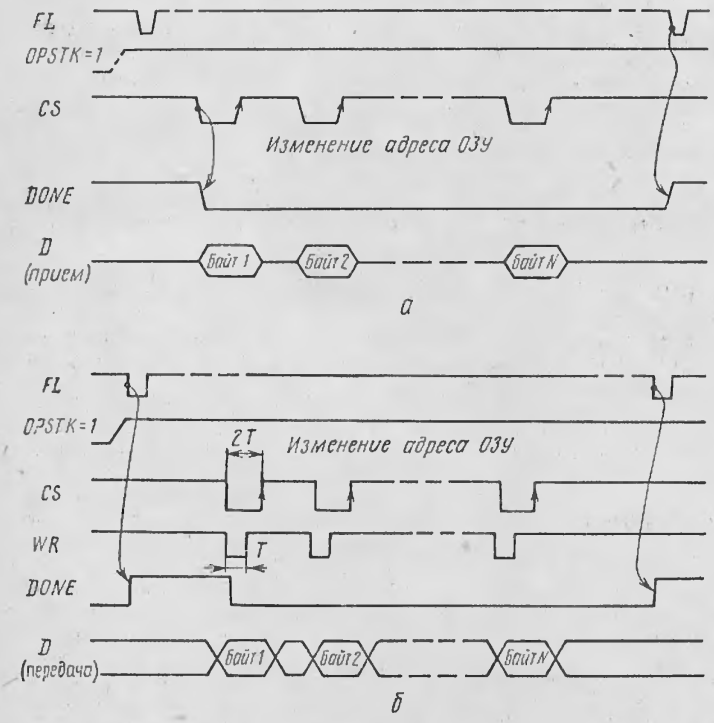


Рис. 6. Временные диаграммы работы микросхемы через ОЗУ/МОД-1 в режимах записи (*а*) и чтения (*б*)

пульсного запроса C9. Сигналы CS и WR специально предназначены для использования в этом случае дополнительного буферного ОЗУ. Временна́я диаграмма формирования страницы данных в режиме «работа с ОЗУ (шиной D)» представлена на рис. 6. При выполнении операции ЗАПИСЬ данные в ОЗУ пользователя необходимо размещать в том порядке, в каком они будут записаны в ЗУ ЦМД (в порядке следования байтов).

Если MOD=0 (работа через внутренний стек микросхемы), то пользователь может опоздать с обменом байта не более чем на три запроса CS в течение формирования одной страницы данных. При этом предназначенные для обмена байты накапливаются в СОЗУ. При операции ЧТЕНИЕ подготовленный к передаче байт присутствует в буфере БФЗ (на шине D, если CS=0) в течение активного сигнала DONE=0. Пользователь сопровождает каждый обмен байта импульсным сигналом OPSTK=0, получив который, СУС продвигает указатель стека со стороны пользователя (изменение адресов следующее: 0→1→2→→3→0→1 и т. д.) и сбрасывает флаг запроса передачи DONE.

Особенности режима «работа через стек»

Режим применяется при работе ФД непосредственно на шину данных ведущего процессора (микропроцессора) без буферного ОЗУ. Сигналы DONE и OPSTK при этом могут быть использованы для реализации режима прямого доступа к памяти. При выполнении операции ЗАПИСЬ перед первым появлением сигналов SH1 и SH2 в стеке должен находиться хотя бы один байт информации, так как БУ во время действия первого сигнала SH1 записывает байт информации в регистр РГ3 из верхушки стека.

Временна́я диаграмма формирования страницы данных в режиме ЗАПИСЬ показана на рис. 7, *а*. До начала формирования страницы ведущий процессор заполняет стек (байты 1...4). Первый переход в активное состояние выхода DONE сигнализирует системе о начале формирования страницы (байт из верхушки стека поступил в регистр РГ3 на обработку). С этого момента ведущий процессор в ответ на каждый запрос DONE выставляет на шине очередной байт данных, сопровождая его стробом OPSTK. Форматор фиксирует данные по положительному перепаду OPSTK. Система ведет подсчет переданных байтов, включая предварительно загруженные, и заканчивает обмен, когда счетчик байтов исчерпан. Приведена одна из возможных ситуаций: полная предварительная загрузка стека (четыре байта) и максимально возможная пауза в обмене данными (пропущены три запроса, подсчет — по сигналу CS).

Временна́я диаграмма формирования

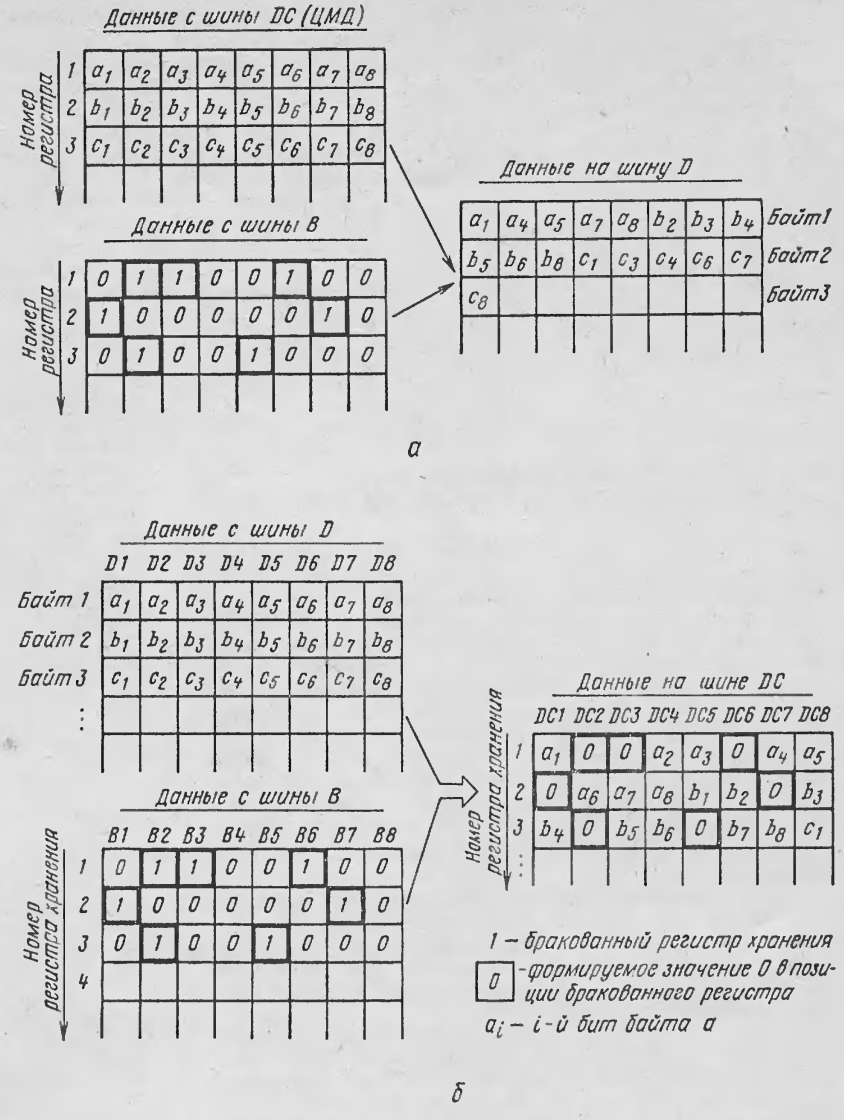


Рис. 8. Принцип формирования массивов информации в режимах чтения (*а*) и записи (*б*)

страницы в режиме ЧТЕНИЕ показана на рис. 7, *б*. В процессе обмена данными система может отстать, например на три байта, поэтому по окончании формирования страницы, если счетчик байтов не исчерпан, система добирает оставшиеся байты из стека, несмотря на то, что запрос DONE уже не вырабатывается.

Формат данных интерфейса ЗУ ЦМД. Шины B и DC

Восемь разрядов шины браков соответствуют восьми параллельно работающим микросхемам ЗУ ЦМД. Значение Лог. 1 на шине B соответствует дефектному регистру хранения СБИС ЗУ ЦМД, а Лог. 0 — годному. При подключении микросхем памяти необходимо соответствие позиций микросхемы СБИС ЗУ ЦМД разрядам шины браков B и шины DC. Если число микросхем меньше восьми, то в позициях отсутствующих микросхем в соответствующих разрядах шины B необходимо поддерживать высокое значение уровня сигнала — Лог. 1, что обеспечит их обход как полностью дефектных. Информация на шине B не должна меняться во время активного сигнала SH1.

Алгоритм формирования информационных массивов форматором данных поясняется на рис. 8. Для эффективного использования СБИС ЗУ ЦМД с учетом того, что имеются дефектные регистры, выбрана стратегия размещения информации в параллельно работающих микросхемах памяти, когда для размещения информационной страницы необходимо иметь суммарное число годных регистров во всех микросхемах. При этом в режиме записи в ЗУ ЦМД (процесс подготовки байта для записи) запись текущей по-

зиции бита байта данных будет задержана до появления первого годного регистра ЗУ ЦМД, который может принадлежать любой из параллельно работающих микросхем памяти (рис. 8, *а*). Очевидно, что информационная емкость страницы ЗУ ЦМД, доступная для пользователя, будет равна

$$\left(\frac{n l - \sum_{i=1}^{n} k_i}{8} \right),$$

где n=1...8 — число параллельно работающих микросхем ЗУ ЦМД; l — число регистров хранения, определяется типом СБИС ЗУ ЦМД, например, для микросхем К1602РЦ2 l=282; k_i — число бракованных регистров i-й микросхемы;) (— операция целочисленного деления.

Микросхема К1806ХМ1-158 выполнена по КМОП-технологии, требует для своей работы один источник питания U=+5 В±10 %, совместима с ТТЛ-логикой и отличается невысоким энергопотреблением — $P_{ст} \leqslant 6$ мВт.

Частота внешнего тактового генератора CLK может достигать 8 МГц. При этом, если используются восемь параллельно работающих микросхем и частота ВМП равна 100 кГц, то CLK=6,4 МГц, если меньше или равно четырем, то CLK может быть снижен до 3,2 МГц.

Телефон: 944-65-33, Москва

Статья поступила 7.07.88

УДК 681.323

В. К. Шмат, Д. В. Кокорев, Н. Н. Усов

КОНТРОЛЛЕРЫ ВНЕШНИХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ НА ЦМД

Минимальный базовый комплект БИС управления (см. предыдущую статью) позволяет разрабатывать системы памяти на ЗУ ЦМД различных типов. Один из вариантов устройства управления для памяти на ЗУ ЦМД емкостью 256 Кбайт, построенной на микросхемах типа К1602РЦ2, успешно используется в микропроцессорной системе на основе МП КР580ВМ80А. В состав устройства управления входит контроллер К1806ВП1-157 и форматор данных К1806ХМ1-158 [1]. Ведущий микропроцессор системы адресуется к микросхемам комплекта как к портам ввода-вывода (два порта для контроллера и один порт для форматора данных). Порт выбирается с помощью дешифратора адреса ДШ и сигналов интерфейса запись-чтение порта: $\overline{\text{ЧтВВ}}$ и $\overline{\text{ЗпВВ}}$, вырабатываемых микропроцессором.

Один порт используется для информационных операций с регистром команд и состояний контроллера (A=0), другой — для доступа к регистровому файлу (A=1), последний — для информационного обмена с ЗУ ЦМД через стек (СОЗУ) форматора. Готовность форматора к обмену подтверждает сигнал ГОТОВ DONE.

Выход на шину данных системы осуществляется через буферные схемы шинных формирователей ШФ типа К589АП16, К589АП26, КР580ВА86, КР580ВА87. Для системы, насчитывающей восемь микросхем ЗУ ЦМД, доменная память представляет собой устройство блочной структуры с прямым доступом. Емкость блока равна 256 байт, адресации — 0...1024. Если применять всего одну микросхему памяти, то емкость устройства будет составлять 1025 блоков×32 байта.

Связь контроллера с микроЭВМ типа «Электроника 60» осуществляется через устройство параллельного обмена типа И2. Для организации программного обмена с контроллером используются следующие регистры интерфейса И2: команд и состояний с адресом 167770 (VKCS); записи (выходной буфер) 167772 (VKWR); чтения (входной буфер) 167774 (VKRD).

При необходимости пользователь может изменять адреса регистров в пределах адресного пространства страницы ввода-вывода микроЭВМ в диапазоне 160000...177776.

Контроллер, использующий в качестве буфера данных стек микросхемы форматора данных

Контроллер содержит микросхемы К1806ВП1-157 (D11) и К1806ХМ1-158 (D12) (рис. 1, 2). Двунаправленные шины данных D1...D8 обеих микросхем через шинные формирователи К589АП16 (D1 и D2) подключены к интерфейсу И2. Формат и назначение битов регистров контроллера приведены на рис. 3. Пользователь имеет доступ к внутренним регистрам микросхемы К1806ВП1-157 и к стеку микросхемы К1806ХМ1-158. Синхронизация устройства производится тактовым генератором, реализованным на микросхемах D3 (К155ЛН1), D 7.1. (К155ТМ2) и кварцевом резонаторе ВР1. На тактовый вход CLK микросхемы D12 поступает тактовый сигнал частотой 6,4 МГц, а на вход G микросхемы D11 — 3,2 МГц. Для сопряжения с шиной браков применены К559ИП2П (D9 и D10), с шиной данных ЦМД — КР580ВА87 (D8). Необходимые для работы СБИС ЗУ ЦМД сигналы управления, формируемые микросхемой К1806ВП1-157, которые поступают на шину управления через формирователи КР580ВА86 (D13... D15):

$\overline{\text{COIL}}$ — разрешение работы формирователей токов продвижения доменов;

$\overline{\text{FA}}$, $\overline{\text{FB}}$, $\overline{\text{FC}}$, $\overline{\text{FD}}$ — четыре импульса, сдвинутых относительно друг друга на четверть периода следования для управления формирователями токов продвижения доменов; последовательность фаз следующая: $\overline{\text{FD}}$, $\overline{\text{FA}}$, $\overline{\text{FB}}$, $\overline{\text{FC}}$, $\overline{\text{FD}}$, $\overline{\text{FA}}$, ..., $\overline{\text{FC}}$, $\overline{\text{FD}}$;

$\overline{\text{PROM}}$ — разрешение работы ПЗУ браков модулей памяти;

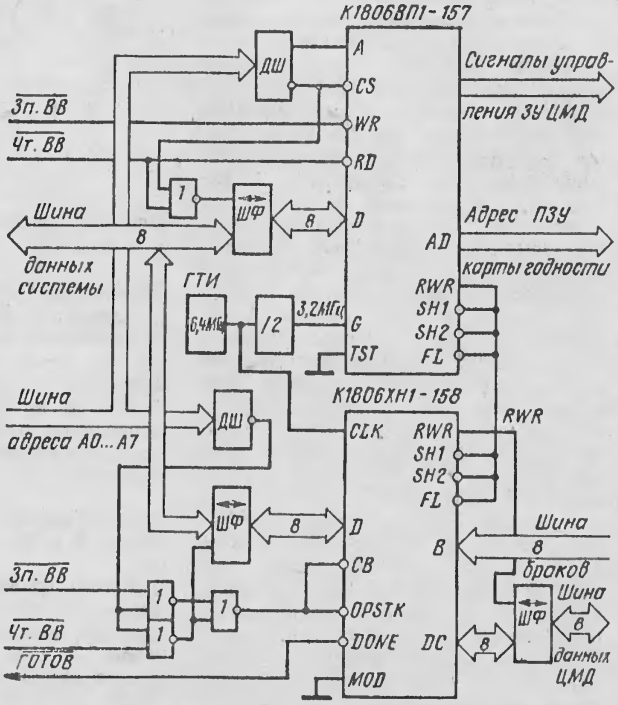


Рис. 1. Устройство управления памятью на ЦМД для микропроцессорных систем

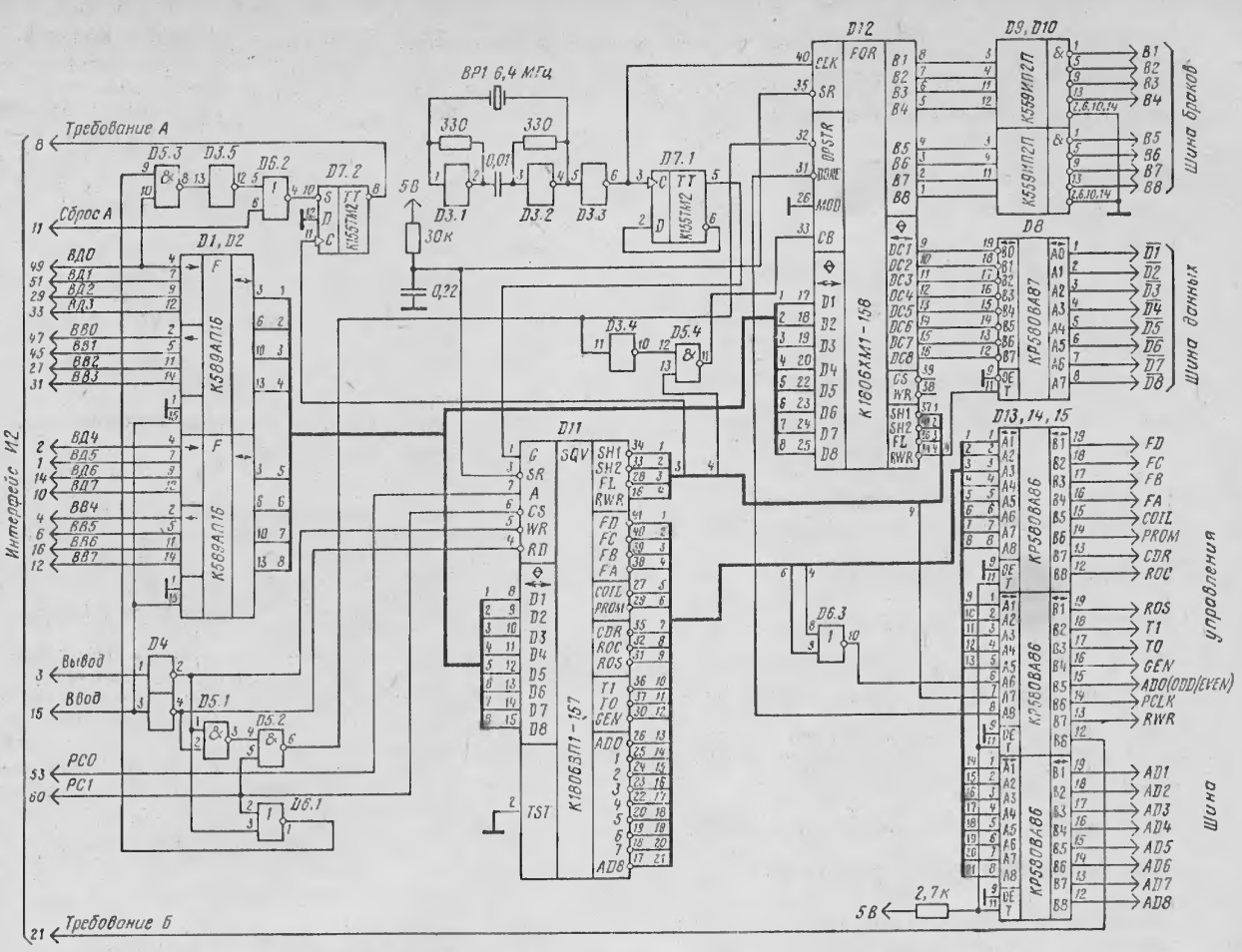


Рис. 2. Принципиальная электрическая схема контроллера

$\overline{ROC}$, $\overline{ROS}$ — сигналы репликации доменов;

$\overline{GEN}$ — сигнал разрешения работы генератора доменов;

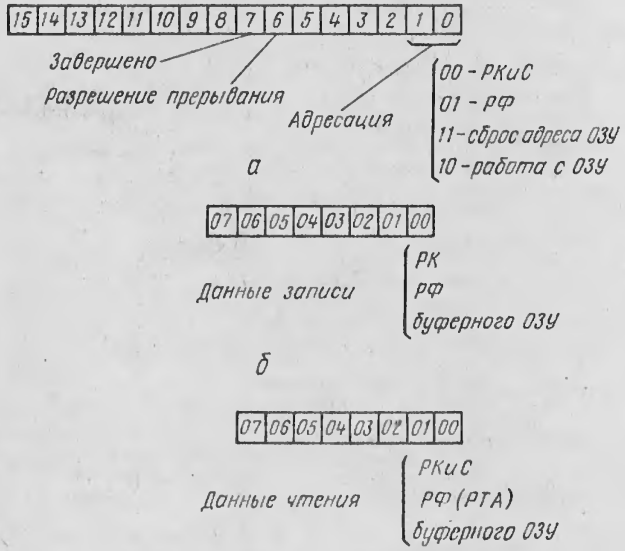


Рис. 3. Форматы регистра команд и состояний VKCS (*а*), записи VKWR (*б*), чтения VKRD (*в*)

$\overline{T0}$, $\overline{T1}$ — переключатель вывода-ввода;

$\overline{CDR}$ — строб усилителя чтения;

ODD/EVEN — разделение каналов чтения СБИС ЗУ ЦМД (используется сигнал AD0);

$\overline{PCLK}$ — продвижение счетчика адреса ПЗУ браков модулей памяти (если ПЗУ не используют счетчик адреса, то работают сигналы AD 0...AD 9);

AD 0...AD 9 — адрес для ПЗУ браков;

RWR — тип операций, выполняемых СБИС ЗУ ЦМД и используемых для переключения приемопередатчиков шины данных выбранного модуля памяти.

Для организации процедур обмена запись-чтение с платой И2 используются сигналы ВВОД (чтение данных процессором ЭВМ), ВЫВОД, РС0, РС1. Логика управления выполнена на D4...D6 серии К155. Внутренняя двунаправленная шина данных контроллера позволяет производить запись-чтение регистров микросхемы D11 или стека микросхемы D12. Основное состояние шины — микросхемы D12 и D11 работают на прием (состояние «отключено»). Для согласования работы микросхем D11 и D12 соответствующие выводы SH1, SH2, FL, RWR, образующие внутренний интерфейс комплекта, соединены между собой. Сигнал RWR используется также для управления шинным формирователем D8 и организации управления шиной микросхемы D12. При выполнении операции ЗАПИСЬ ЦМД шина микросхемы D12 работает на прием информации, при операции ЧТЕНИЕ ЦМД для управления шиной используется управляющий вход СВ и вентили ИС D3.4 и D5.4. Вход

```
;*************************************
;*КОНТРОЛЛЕР НА БИС К1806ВП1-157 И БИС *
;*К1806ХМ1-158 С РАБОТОЙ ЧЕРЕЗ ВНУТРЕН-*
;*       НИЙ СТЕК ФОРМАТОРА            *
;*************************************


;*************************************
;*   ПРИМЕРЫ   ПРОГРАММ ОБСЛУЖИВАНИЯ   *
;*************************************

;АДРЕСА РЕГИСТРОВ КОНТРОЛЛЕРА
VKCS    =167770 ;РЕГИСТР КОМАНД И СОСТОЯНИЙ
VKWR    =VKCS+2 ;РЕГИСТР ЗАПИСИ
VKRD    =VKCS+4 ;РЕГИСТР ЧТЕНИЯ

;АДРЕСАЦИЯ К БИС КОМПЛЕКТА

PC1     =0      ;АДРЕСАЦИЯ К РЕГИСТРАМ
                ;К1806ВП1-157
PC0     =0      ;PKS-РЕГИСТР КОМАНД И СОСТОЯНИЙ
PC0     =1      ;РФ -РЕГИСТРОВЫЙ ФАЙЛ


PC1     =1      ;АДРЕСАЦИЯ К СТЕКУ ФОРМАТОРА
                ;К1806ХМ1-158

;*************************************
;*ЧТЕНИЕ РТА-РЕГИСТРА ТЕКУЩЕГО АДРЕСА*
;*      К1806ВП1-157                 *
;*************************************

CLR     @#VKCS          ;УСТАНОВИТЬ АДРЕСАЦИЮ
                        ;К PKS
MOV     #40,@#VKWR      ;УСТАНОВИТЬ УКАЗАТЕЛЬ
                        ;LSB РТА
INC     @#VKCS          ;АДРЕСАЦИЯ К РФ
MOVB    @#VKRD,PTA      ;МЛАДШИЙ БАЙТ РТА
CLR     @#VKCS          ;АДРЕСАЦИЯ К PKS
MOV     #140,@#VKWR     ;УСТАНОВИТЬ УКАЗАТЕЛЬ
                        ;К MSB РТА
INC     @#VKCS          ;УСТАНОВИТЬ АДРЕСАЦИЮ
                        ;К РФ
MOVB    @#VKRD,PTA+1    ;MSB (СТАРШИЙ БАЙТ)
                        ;РТА
RETURN
PTA:    .WORD 0         ;ЗНАЧЕНИЕ РТА
PAП:    .WORD 0         ;ЗНАЧЕНИЕ РАП

;*************************************
;*      ЗАПИСЬ   РТА                 *
;*************************************

CLR     @#VKCS          ;УСТАНОВИТЬ
MOV     #40,@#VKWR      ;АДРЕСАЦИЮ К
INC     @#VKCS          ;LSB РТА
MOVB    PTA,@#VKWR      ;ЗАПИСЬ LSB РТА
CLR     @#VKCS
MOV     #140,@#VKWR     ;УСТАНОВИТЬ УКАЗАТЕЛЬ
                        ;НА MSB РТА
INC     @#VKCS
MOVB    PTA+1,@#VKWR    ;СТАРШИЙ БАЙТ РТА
RETURN

;*************************************
;*      ЗАПИСЬ РАП                   *
;*************************************

CLR     @#VKCS          ;УСТАНОВИТЬ
MOV     #240,@#VKWR     ;АДРЕСАЦИЮ К
INC     @#VKCS          ;LSB РАП
MOVB    РАП,@#VKWR      ;LSB РАП
                        ;ЗАПИСАН В РФ
CLR     @#VKCS          ;УСТАНОВИТЬ АДРЕСАЦИЮ
MOV     #340,@#VKWR     ;К MSB РАП
INC     @#VKCS
MOVB    РАП+1,@#VKWR    ;ЗАПИСАТЬ MSB РАП
RETURN

;*************************************
;* ЧТЕНИЕ И АНАЛИЗ РС К1806ВП1-157 *
;*************************************
```

а

```
        CLR     @#VKCS          ;УСТАНОВИТЬ АДРЕСАЦИЮ
                                ;К РС К1806ВП1-157
        MOVB    @#VKRD,R0
        TSTB    R0              ;ЕСТЬ ОШИБКА?
        BMI     ERROR           ;ДА.ПЕРЕЙТИ НА ПОДПРОГРАММУ
                                ;ОБРАБОТКИ ОШИБКИ
        BITB    #100,R0         ;ЦМД РАБОТАЕТ?
                                ;COIL=0
        BEQ     COIL            ;ДА
        CLR     FL              ;ЦМД НЕ РАБОТАЕТ
        RETURN
COIL:   MOVB    #200,FL         ;УСТАНОВИТЬ В ФЛАГЕ
                                ;БИТ COIL
        BITB    #40,R0          ;ФАЗА ДАННЫХ?
                                ;PROM=0
        BEQ     PROM            ;ДА
        RETURN
PROM:   BISB    #100,FL         ;УСТАНОВИТЬ В ФЛАГЕ
                                ;БИТ PROM
        RETURN
FL:     .WORD   0               ;ФЛАГ СОСТОЯНИЯ

ERROR:  ТОЧКА ВХОДА НА П/ПРОГРАММУ
        ОБРАБОТКИ ОШИБКИ

        ;*************************************
        ;* ЗАДАНИЕ ОПЕРАЦИИ КОНТРОЛЛЕРУ  *
        ;*       К1806ВП1-157            *
        ;*************************************

        CLR     @#VKCS          ;УСТАНОВИТЬ АДРЕСАЦИЮ
                                ;К PKS
        MOVB    (PC)+,@#VKWR    ;ПЕРЕСЛАТЬ КОД КОМАНДЫ
KOD:    .WORD   200             ;ЗДЕСЬ КОД КОМАНДЫ
        TSTB    @#VKRD          ;КОД КОМАНДЫ ПРАВИЛЬНЫЙ?
        BMI     ERRCR           ;НЕДОПУСТИМЫЙ КОД
        RETURN
ERROR:  ВХОД НА ПОДПРОГРАММУ
        АНАЛИЗА ОШИБКИ:
        1.ДОПУСТИМЫЙ КОД КОМАНДЫ?
        2.ДОПУСТИМОЕ ЗНАЧЕНИЕ АДРЕСА?
        ;*************************************
        ;*  ЗАПИСЬ СТРАНИЦЫ ЧЕРЕЗ СТЕК  *
        ;*       К1806ХМ1-158           *
        ;*************************************

;R0     -АДРЕС МАССИВА ЗАПИСИ
;R1     -СЧЕТЧИК БАЙТОВ
;       ВНИМАНИЕ:УЖЕ БЫЛА ВЫДАНА КОМАНДА
;               ЗАПИСИ

        MOV     #VKCS,R2
        MOV     #VKWR,R4
        MOV     #256.,R1        ;ДЛИНА СТРАНИЦЫ
                                ;РАВНА 256 БАЙТ
        MOV     #2,(2)          ;УСТАНОВИТЬ АДРЕСАЦИЮ
                                ;К СТЕКУ ФОРМАТОРА
                                ;К1806ХМ1-158
        MOVB    (0)+,(4)        ;ПЕРЕСЛАТЬ ПЕРВЫЕ
        MOVB    (0)+,(4)        ;ЧЕТЫРЕ БАЙТА В СТЕК
        MOVB    (0)+,(4)        ;ФОРМАТОРА К1806ХМ1-158
        MOVB    (0)+,(4)
1Ж:     TST     (2)             ;СТЕК ГОТОВ К ПРИЕМУ?
        BMI     1Ж              ;НЕТ,ЖДАТЬ
        MOVB    (0)+,(4)        ;ПОДКАЧКА СТЕКА
        SOB     R1,1Ж           ;ИСЧЕРПАН СЧЕТЧИК
                                ;БАЙТ?
        RETURN                  ;ДА,ЗАВЕРШИТЬ.

        ;*************************************
        ;* ЧТЕНИЕ СТРАНИЦЫ ЧЕРЕЗ СТЕК   *
        ;* ФОРМАТОРА К1806ХМ1-158       *
        ;*************************************

;R0     -АДРЕС МАССИВА В ОЗУ ЭВМ
;R1     -СЧЕТЧИК БАЙТОВ СТРАНИЦЫ

        MOV     #VKCS,R2
        MOV     #VKRD,R4
        MOV     #256.,R1        ;СЧЕТЧИК БАЙТОВ
1Ж:     TST     (2)             ;ГОТОВ БАЙТ В СТЕКЕ?
        BMI     1Ж              ;НЕТ,ЖДАТЬ
        MOVB    (4),(0)+        ;ДА,ПЕРЕСЛАТЬ В ОЗУ ЭВМ
        SOB     R1,1Ж           ;ВСЯ СТРАНИЦА?
        RETURN                  ;ДА
```

б

Рис. 4. Программы обслуживания контроллера с работой через внутренний стек форматора (*а*) и через буферное ОЗУ (*б*)

```
                ********************************
                *КОНТРОЛЛЕР НА БИС К1806ВП1-157 *
                *И К1806ХМ1-158 С РАБОТОЙ ЧЕРЕЗ *
                *     БУФЕРНОЕ ОЗУ К537РУ8      *
                * ПРИМЕРЫ ПРОГРАММ ОБСЛУЖИВАНИЯ *
                ********************************

                ;АДРЕСА РЕГИСТРОВ КОНТРОЛЛЕРА
                VKCS    =167770
                VKWR    =VKCS+2
                VKRD    =VKCS+4

                ;НАЗНАЧЕНИЕ БИТ РС0 И РС1 VKCS
;               РС0 = 0 -АДРЕСАЦИЯ К РЕГИСТУ
;               РС1 = 0 -РКS К1806ВП1-157


;               РС0 = 1 -АДРЕСАЦИЯ К РЕГИСТРОВОМУ
;               РС1 = 0 -ФАЙЛУ РФ К1806ВП1-157

;               РС0 = РС1 = 1---СБРОС СЧЕТЧИКА
;                               АДРЕСА БУФЕРНОГО
;                               ОЗУ

;               РС0 = 0  -РАБОТА С БУФЕРНЫМ
;               РС1 = 1  -ОЗУ КОНТРОЛЛЕРА

;               ПРОГРАММЫ ЧТЕНИЯ И ЗАПИСИ РЕГИСТРОВ
;               МИКРОСХЕМЫ К1806ВП1-157 АНАЛОГИЧНЫ
;               ПРИВЕДЕННЫМ ДЛЯ ВАРИАНТА КОНТРОЛЛЕРА
;               РАБОТАЮЩЕГО ЧЕРЕЗ СТЕК ФОРМАТОРА


                ********************************
                * ЗАПИСЬ СТРАНИЦЫ БУФЕРНОГО ОЗУ *
                ********************************

;               R0 - АДРЕС МАССИВА ОЗУ ЭВМ
;               R1 - СЧЕТЧИК БАЙТОВ СТРАНИЦЫ
;               R5 - КОНТРОЛЬНАЯ СУММА
;               КОМАНДА "ЗАПИСЬ СТРАНИЦЫ"
;               ВЫДАЕТСЯ ПОСЛЕ ЗАПИСИ ДАННЫХ
;               В БУФЕРНОЕ ОЗУ КОНТРОЛЛЕРА

        WRITE:  MOV     #VKCS,R2
                MOV     #VKWR,R4
                MOV     #256.,R1        ;УСТАНОВКА СЧЕТЧИКА
                                        ;БАЙТ СТРАНИЦЫ
                CLR     R5              ;КОНТРОЛЬНАЯ
                                        ;СУММА =  0
                MOV     #3,(2)          ;СБРОС СЧЕТЧИКА
                                        ;БУФЕРНОГО ОЗУ
                                        ;КОНТРОЛЛЕРА
                DEC     (2)             ;УСТАНОВИТЬ
                                        ;РЕЖИМ РАБОТЫ С ОЗУ
MOVV:           MOVB    (0)+,R3         ;БАЙТ ИЗ ОЗУ ЭВМ
                BEQ     1$              ;ЕСЛИ 0,ТО ПЕРЕСЛАТЬ
                                        ;ЕСЛИ НЕ 0,ТО
                BIC     #177400,R3      ;ФОРМИРОВАНИЕ КОНТРОЛЬНОЙ
                ADD     R3,R5           ;СУММЫ
1$:             MOVB    R3,(4)          ;ПЕРЕСЫЛКА БАЙТА В ОЗУ
                                        ;КОНТРОЛЛЕРА
                SOB     R1,MOVV         ;ЗАВЕРШЕНО?
                MOVB    R5,(4)          ;1-ЫЙ БАЙТ КОНТРОЛЬНОЙ
                                        ;СУММЫ
                SWAB    R5
                MOVB    R5,(4)          ;ВТОРОЙ БАЙТ КОНТ.СУММЫ
                MOV     #3,(2)          ;СБРОС УКАЗАТЕЛЯ АДРЕСА
                                        ;БУФЕРНОГО ОЗУ
                RETURN

                ********************************
                * ЧТЕНИЕ ДАННЫХ ИЗ БУФЕРНОГО ОЗУ *
                *          КОНТРОЛЛЕРА           *
                ********************************

;               R0 - АДРЕС ОБЛАСТИ ПАМЯТИ В ОЗУ ЭВМ
;               R1 - СЧЕТЧИК БАЙТОВ СТРАНИЦЫ
;               R5 - РЕГИСТР КОНТРОЛЬНОЙ СУММЫ
;               ЧТЕНИЕ ИЗ БУФЕРНОГО ОЗУ ПОСЛЕ
;               ЗАВЕРШЕНИЯ ОПЕРАЦИИ ЧТЕНИЯ ИЗ ЦМД

        READ:   MOV     #VKCS,R2
                MOV     #VKRD,R4
                MOV     #256.,R1        ;СЧЕТЧИК БАЙТОВ
                CLR     R5              ;КОНТРОЛЬНАЯ
                                        ;СУММА = 0
                MOV     #3,(2)          ;СБРОС АДРЕСА
                                        ;БУФЕРНОГО ОЗУ
                MOV     #2,(2)          ;УСТАНОВИТЬ РЕЖИМ
                                        ;РАБОТЫ С ОЗУ
RD:             MOVB    (4),R3          ;БАЙТ ИЗ ОЗУ
                                        ;КОНТРОЛЛЕРА
                BEQ     1$
                BIC     #177400,R3      ;ФОРМИРОВАНИЕ
                ADD     R3,R5           ;КОНТРОЛЬНОЙ
                                        ;СУММЫ
1$:             MOVB    R3,(0)+         ;БАЙТ В ОЗУ ЭВМ
                SOB     R1,RD
                MOVB    (4),R3          ;ПЕРВЫЙ БАЙТ КОНТРОЛЬНОЙ
                                        ;СУММЫ
                BIC     #177400,R3
                MOVB    (4),R1          ;ВТОРОЙ БАЙТ КОНТРОЛЬНОЙ
                BIC     #177400,R1      ;СУММЫ
                SWAB    R1
                BIS     R1,R3           ;КОНТРОЛЬНАЯ СУММА
                XOR     R5,R3           ;ПРОВЕРКА КОНТРОЛЬНОЙ СУММЫ
                BNE     ERREAD          ;ОШИБКА,ПЕРЕХОД НА
                                        ;ПРОГРАММУ ОБРАБОТКИ ОШИБКИ
                RETURN
ERREAD:         ТОЧКА ВХОДА В ПРОГРАММУ
                ОБРАБОТКИ ОШИБКИ
```

Рис. 4, *б*

MOD микросхемы D12 подключен к шине нулевого потенциала устройства для задания режима «работа через стек». Входы SR начальной установки обеих микросхем подключены к RC-цепочке, которая обеспечивает формирование сигнала начальной установки микросхем при включении питания, переводящей их в детерминированное состояние. При этом происходит сброс внутренних регистров К1806ВП1-157 и установка К1806ХМ1-158 в состояние готовности к выполнению операции, задаваемой по входу RWR. Для организации программного обмена с процессором ЭВМ используются сигналы OPSTK и DONE К1806ХМ1-158. Алгоритмы выполнения операций следующие:

ЧТЕНИЕ. 1. Задание адреса в регистре адреса поиска РАП контроллера К1806ВП1-157 (если необходимо).

2. Задание операции ЧТЕНИЕ записью кода команды в РК микросхемы D11.

3. Ожидание готовности форматора к обмену опросом флага $\overline{\text{DONE}}$ («Требование Б»).

4. Пересылка байта из стека форматора в область памяти ОЗУ ЭВМ.

5. Повторение п. 3 и 4 до исчерпания счетчика байтов обмена.

6. Проверка завершения операции чтения блока при многоблочных операциях по РС контроллера К1806ВП1-157 или завершения операции по прерыванию или опрос флага «Требование А» при одностраничных операциях.

7. Для многоблочных операций — повторение п. 3, 4, 5, 6 до исчерпания счетчика количества блоков обмена. Если исчерпан счетчик блоков (страниц) обмена, то запись кода команды СТОП=200_8 в РК контроллера К1806ВП1-157 и проверка завершения операции по состоянию РС контроллера.

ЗАПИСЬ. 1. Задание адреса в РАП (если необходимо).

2. Задание операции ЗАПИСЬ через РК микросхемы D11.

3. Загрузка четырех байтов в стек форматора данных.

4. Ожидание готовности форматора данных к обмену по опросу флага «Требование Б» $\overline{\text{DONE}}$.

5. Пересылка байта из области памяти ОЗУ ЭВМ в стек форматора данных.

6. Повторить п. 4 и 5 до исчерпания счетчика байт обмена страницы.

7. Если операция одностраничная, то завершить выполнение проверкой флага «Требование А».

8. Если операция многостраничная, то проверить счетчик страниц после его декремента. Если он исчерпан, то выдать через РК команду СТОП=200_8 и ожидать завершения

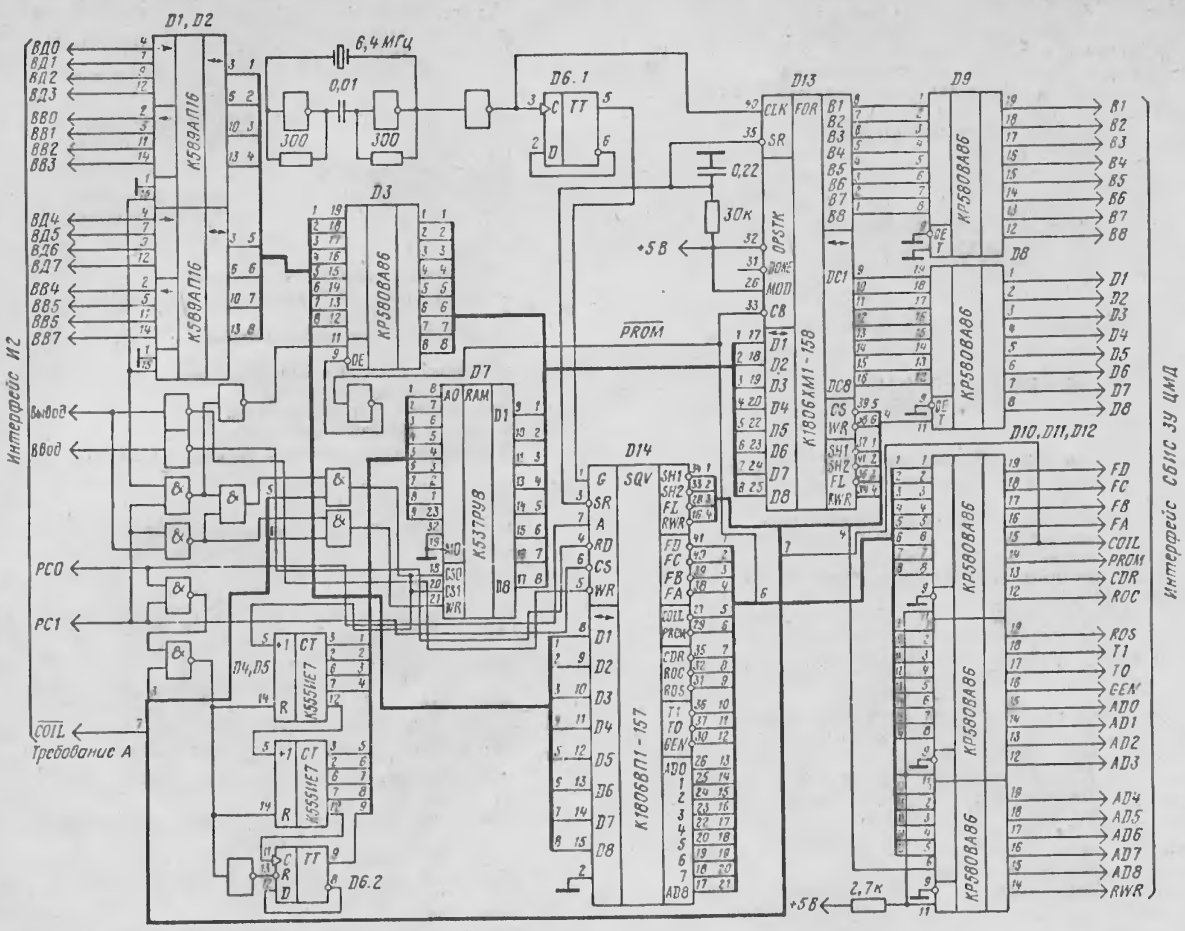


Рис. 5. Принципиальная электрическая схема контроллера, использующего буферное ОЗУ

операции по РС; если счетчик не исчерпан, то ожидать окончания фазы данных — бит PROM=1 в РС микросхемы D11.

9. Четырехкратная загрузка первого байта данных следующей страницы в стек форматора данных.

10. Ожидание готовности форматора данных к обмену по опросу флага «Требование Б».

11. Запись в стек форматора данных следующих трех байтов из области данных ОЗУ ЭВМ, уменьшение счетчика байтов страницы.

12. Перейти к выполнению п. 4.

Замечание. п. 10 допускает следующую редакцию — «Ожидание фазы данных следующей страницы», например по прерыванию, по сигналу $\overline{\text{PROM}}$ микросхемы D11 или по опросу флага РС: бит PROM=0. В п. 9 введена четырехкратная загрузка стека, так как его положение (адрес) не известно, тем более что по окончании записи страницы указатель стека будет установлен в начальное положение. Полное описание команд контроллера К1806ВП1-157 приведено в работе [2]. На рис. 4 приводятся тексты программ обслуживания контроллера, используя которые, можно модифицировать драйвер устройства. Допускается объединять код команды с записью указателя регистрового файла, если установлен бит разрешения записи указателя регистрового файла. Следовательно, каждая команда, код которой указан в табл. 5 работы [2], может быть модифицирована с учетом трех старших битов, не влияющих на определение типа операции.

Контроллер, использующий буферное ОЗУ (рис. 5)

Для выполнения информационных операций используется буферное ОЗУ К537РУ8 (D7). Управление адресом ОЗУ выполнено на микросхемах К555ИЕ7 (D4, D5) и триггере К155ТМ2 (D6.2). Управление памятью К537РУ8 осуществляется микросхемой форматора — выводы CS, WR или от интерфейса И2. Буферное ОЗУ функционирует в режиме стека FIFO. Выводы OPSTK и МОД форматора D13 подключены к шине 5 В. Входы-выходы данных микросхемы К537РУ8 соединены с шиной данных D1... D8 форматора D13 и входами-выходами шинного формирователя КР580ВА86 (D3). Управление шиной D форматора осуществляет микросхема контроллера К1806ВП1-157 (D14). Для этого ее выход 29 PROM соединен с входом СВ (вывод 33 микросхемы форматора D13) и через инвертор с управляющим входом 9 микросхемы D3. Следовательно, при активном сигнале PROM=0 входы-выходы микросхемы D3 переводятся в состояние «отключено» и управление шиной данных «ОЗУ-форматор» осуществляет сам форматор.

Назначение битов и формат регистров контроллера приведены на рис. 6. В зависимости от значения битов РС0 и РС1 регистра команд и состояний VKCS возможно выполнение следующих операций:

РС1=0 — разрешен обмен с регистрами контроллера К1806ВП1-157, если РС0=0, то адресуется РКиС, если РС0=1, один из байтов регистрового файла;

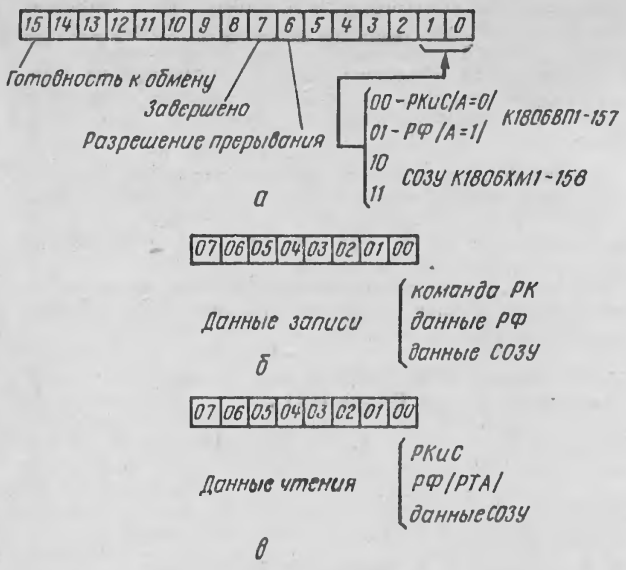


Рис. 6. Форматы регистров команд и состояний VKCS *(а)*, записи VKWR *(б)*, чтения VKPD *(в)* контроллера, использующего буферное ОЗУ

PC1=1 и PC0=1 — сброс указателя адреса буферного ОЗУ;

PC1=1 и PC0=0 — разрешена работа с буферным ОЗУ в режиме стека.

Во время фазы данных при работе СБИС ЗУ ЦМД (PROM=0) запрещена связь пользователя с буферным ОЗУ. Алгоритмы выполнения операций записи и чтения страницы при этом будут следующие:

ЧТЕНИЕ. 1. Задание адреса в регистре адреса поиска РАП (если необходимо).

2. Сброс указателя адреса буферного ОЗУ.

3. Задание операции ЧТЕНИЕ записью кода команды в РК микросхемы контроллера D14.

4. Ожидание завершения операции по опросу флага «Требование А» или по прерыванию.

5. Чтение буферного ОЗУ:

5.1. Установить PC1=1 и PC0=0 в WCS.

5.2. Переслать байт из ОЗУ в память ЭВМ.

5.3. Если счетчик байтов не исчерпан (после его декремента), то повторить п. 5.2.

Параллельно с пересылкой байта процессор ЭВМ может вычислять значение контрольной суммы.

6. Завершение операции.

ЗАПИСЬ. 1. Запись буферного ОЗУ.

1.1. Сброс указателя адреса ОЗУ контроллера.

1.2. Пересылка массива данных в буферное ОЗУ, включая и контрольную сумму.

1.3. Сброс указателя адреса буферного ОЗУ.

2. Задание адреса в РАП (если необходимо).

3. Задание операции ЗАПИСЬ через РК микросхемы D14.

4. Ожидание завершения операции по опросу флага «Требование А» или по прерыванию программы.

Используя текст программы, приведенный на рис. 4, *б*, можно модифицировать драйвер [1].

Телефон: 944-65-33, Москва

ЛИТЕРАТУРА

1. Шмат В. К. Контроллер на основе СБИС ЗУ ЦМД для измерительно-вычислительных комплексов // Микропроцессорные средства и системы.— 1987.— № 6.— С. 9.

2. Шмат В. К. Специализированные БИС управления для СБИС ЗУ ЦМД // Микропроцессорные средства и системы.— 1987.— № 6.— С. 3.

Статья поступила 7.07.88

УДК 681.327.06(045)

К. Ф. Сарычев, Ю. А. Окрочков

ПАКЕТ ПРОГРАММ В ПОМОЩЬ РАЗРАБОТЧИКУ КОРРЕКТИРУЮЩИХ КОДОВ ДЛЯ ЦМД ЗУ

Пакет содержит программы: перевода элементов поля из степенного представления в многочисленное, проверки периода образующего многочлена кода g(x) и статистического моделирования надежности запоминающего устройства на цилиндрических магнитных доменах (ЦМД ЗУ). Программы написаны на языке Turbo-Basic и предназначены для ПЭВМ типа «Правец 16», ЕС 1840, «Искра 1030», «Нейрон И66».

При проектировании устройств декодирования циклических корректирующих кодов для ЗУ часто требуется укорачивать код путем умножения считанного из ЗУ кодового слова на x^v по модулю g(x), где v — число отбрасываемых разрядов кода. При больших значениях v выполнить перевод x^v в многочленное представление вручную практически невозможно. Для этой цели написана программа, которая позволяет также определить период многочлена g(x).

Исходные данные для программы — коэффициенты многочлена и степень элемента поля; выходные данные — коэффициенты многочленного представления элемента поля и значение периода многочлена.

Принцип работы программы основан на имитационном моделировании регистра сдвига с обратными связями, определяемыми коэффициентами g(x). Максимальная степень образующего многочлена — 256, элемента поля — 32767.

Программа статистического моделирования надежности ЦМД ЗУ обеспечивает определение показателей: математического ожидания и среднего квадратического отклонения наработки ЗУ до отказа, числа накопленных сбоев (отказов) элементов ЗУ, плотности распределения наработки ЗУ до отказа и вероятности безотказной работы ЗУ в 100 точках.

Исходные данные для программы: виды и интенсивности сбоев и отказов элементов ЦМД ЗУ, емкость и организация ЗУ, интервал времени между соседними профилактиками, погрешность моделирования и метод повышения надежности. Моделируются следующие методы повышения надежности ЦМД ЗУ: контроль по четности, контрольное суммирование, коды Хэмминга, Файра, а также БЧХ и повторное считывание при обнаружении неисправимой ошибки. Для ЦМД ЗУ с коррекцией ошибок реализован способ ускорения моделирования, основанный на учете характера воздействий на ЗУ различных видов сбоев.

Объем программ: перевода элементов поля и проверки периода многочлена — 142, статистического моделирования надежности — 765 операторов языка Basic.

Время перевода элемента поля степени 1516 для кода Файра с образующим многочленом степени 14 и определения периода g(x) — не более 5 мин на ПЭВМ «Правец-16», время моделирования надежности ЦМД ЗУ емкостью 0,5 Мбайт на той же ЭВМ с использованием ускорения — 5—10 мин (при числе реализаций моделирования около 1000).

Телефон 362-77-03, Москва

Сообщение поступило 25.11.88

УДК 681.3

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ КОНТРОЛЯ ЦМД-МИКРОСХЕМ И ЦМД ЗУ*

Промышленное освоение запоминающих устройств на цилиндрических магнитных доменах (ЦМД ЗУ) выдвинуло проблему создания автоматизированных систем их контроля (АСК). К этой проблеме непосредственно относится и задача входного контроля ЦМД-микросхем (ЦМД МС), как основного компонента накопителя ЦМД ЗУ. Разработанные в Институте электронных управляющих машин АСК «Поиск 1К» и «Мультитест 1К» предлагают широкий набор возможностей для контроля и испытаний ЦМД МС и ЦМД ЗУ и решают эти задачи.

Основой АСК, разработанных в ИНЭУМе, являются микроЭВМ СМ1800, СМ1300, «Электроника 60» и любые другие совместимые с ними микроЭВМ.

Опыт показывает необходимость контроля ЦМД МС, модуля накопителя ЦМД ЗУ и комплексной проверки работоспособности ЦМД ЗУ.

Задача тестирования ЦМД МС — проверка работоспособности в диапазоне параметров ТУ.

«Поиск 1К» включает в себя следующие блоки элементов: сопряжения с интерфейсом И41, ОШ или МПИ; контроллера для управления записью и считыванием информации в ЦМД МС; программируемых формирователей функциональных импульсов. Тестовое программное обеспечение реализует режимы:

измерения параметров функциональных узлов проверяемой ЦМД МС по допустимому диапазону; изменения амплитуды, фазы и длительности функциональных импульсов, и временной диаграммы работы ЦМД МС;

записи-считывания в ЦМД МС различных кодовых последовательностей (однократное или многократное);

диагностики, анализа ошибок и определения карты расположения дефектных накопительных регистров в ЦМД МС;

документирование результатов контроля.

Информационная емкость испытываемых ЦМД МС — до 4 Мбит.

Задача тестирования модуля накопителя ЦМД ЗУ — это отбор совместимых по заданным параметрам комплектов ЦМД МС, установленных на один модуль; проверка правильности функционирования модулей; определение карты дефектов (КД) накопительных регистров и подготовка файла для прожига ПЗУ КД с выводом информации на АЦПУ (гибкий диск, дисплей).

«Мультитест 1К» содержит БЭ диагностического контроллера (контроллер самого ЦМД ЗУ в тестовом режиме); БЭ модуля накопителя, укомплектованный колодками для установки от четырех до восьми ЦМД МС различных типов; адаптеры связи контроллера с интерфейсами МПИ или ОШ.

Тестовое программное обеспечение реализует режимы:

настройки на работу с определенным типом ЦМД МС;

формирования различных тестовых последовательностей и выполнения тестов в режиме диалога или автоматическом режиме;

диагностики и документирования карты дефектных накопительных регистров каждой из ЦМД МС, установленных на модуле накопителя, в том числе подготовки файла для прожига ПЗУ карты дефектности.

Гарантия работоспособности модулей накопителя — выполнение совокупности тестов, особенно теста, задающего режим «теплового удара» (работа ЦМД МС при минимальной скважности обращения). Время тестирования модуля накопителя с четырьмя ЦМД МС — около 3 ч (для СМ1800) и около 1,5 ч для 16-разрядных ЭВМ.

Программное обеспечение АСК «Поиск 1К» и «Мультитест 1К» для СМ1800 написано на языке БЕЙСИК, СМ1300 и микроЭВМ «Электроника 60» — на языке Паскаль.

АСК позволяет выполнить контроль накопителя на основе разработанных в настоящее время отечественных ЦМД МС: К1602РЦ2, К1605РЦ1, КРЦ080101, К1602РЦ3, КРЦ100104.

Заключительный этап испытаний предполагает контроль совместной работы модулей накопителя с контроллером ЦМД ЗУ. ЦМД ЗУ проверяются по общей методике электромеханических ЗУ, но при максимальной интенсивности поступления запросов на доступ.

Применение разработанных программно-аппаратных тестовых средств обеспечивает изготовление ЦМД ЗУ с вероятностью сбоя 10^{-11} 1/бит.

117812, Москва, ГСП, ИНЭУМ, тел. 135-60-07
(Красовский Виктор Евгеньевич)

* Разработку выполнил коллектив авторов:
В. К. Раев, В. Е. Красовский, С. О. Кузнецов, А. Е. Шотов, Т. А. Бедертдинов, А. Б. Белькевич, А. А. Ланко, О. В. Матвеев, Н. В. Попко, И. Г. Федоров

УДК 621.327

В. В. Лангуев, С. А. Болотов, А. И. Ольшак

КОМПЛЕКТ БИС СЕРИИ КМ1823 ДЛЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ УПРАВЛЕНИЯ ДВИГАТЕЛЯМИ ВНУТРЕННЕГО СГОРАНИЯ

В состав комплекта БИС серии КМ1823, предназначенного для построения микропроцессорных систем управления двигателями внутреннего сгорания с искровым зажиганием, входят следующие микросхемы:

КМ1823ВУ1 — процессор;
КМ1823ВВ1 — устройство ввода-вывода (УВВ);
КМ1823АГ1 — формирователь импульсов зажигания (ФИЗ);
КМ1823 РЕ1 — постоянное запоминающее устройство (ПЗУ).

Данный комплект позволяет создавать электронные блоки (контроллеры) микропроцессорных систем, обеспечивающие управление углом зажигания в зависимости от таких параметров двигателя, как частота вращения коленчатого вала, нагрузка, температура охлаждающей жидкости, а также топливоподачей в режиме принудительного холостого хода.

Процессор КМ1823ВУ1 предназначен для вычисления угла зажигания как функции частоты вращения коленчатого вала двигателя внутреннего сгорания от угла его поворота; формирования импульсов, определяющих момент зажигания, а также сигнала управления электромагнитным клапаном экономайзера принудительного холостого хода. Для синхронной работы процессора с двигателем используется сигнал датчика начала отсчета, а для вычисления углов зажигания и поворота коленчатого вала — сигнал датчика угловых импульсов. Эти сигналы предварительно обрабатываются устройством ввода-вывода КМ1823ВВ1, с выходов которого они подаются в процессор. Информация об изменении угла зажигания в зависимости от частоты вращения коленчатого вала, программа управления электромагнитным клапаном экономайзера принудительного холостого хода и данные, характеризующие тип автомобиля, хранятся во внешнем ПЗУ. Процессор имеет встроенный генератор тактовых импульсов, 8-разрядную шину данных и 6-разрядную адресную шину и работает по жесткому алгоритму, ориентированному на решение конкретной задачи.

Устройство ввода-вывода КМ1823ВВ1 предварительно обрабатывает входную информацию, поступающую от датчиков, и формирует сигналы управления для внутренних

схем электронного блока управления и исполнительных устройств системы управления двигателем. Устройство преобразует входные сигналы начала отсчета и угловых импульсов (в виде, удобном процессору для обработки); формирует сигналы управления преобразователем «Напряжение-время» (входящим в АЦП, работающим по методу двойного интегрирования), 6-разрядный код адреса страницы ПЗУ, сигнал блокировки зажигания при превышении частоты вращения коленчатого вала двигателя заданного значения, сигналы управления одно- или двухканальным коммутатором первичной цепи катушки зажигания; обеспечивает резервный режим работы контроллера при отказе АЦП. Микросхема программируется информацией, считываемой из внешнего ПЗУ.

Формирователь импульсов зажигания КМ1823АГ1 преобразует последовательность импульсов, определяющих момент зажигания, в сигнал с постоянной скважностью, необходимый для управления одно- или двухканальной катушкой зажигания, а также формирует резервные импульсы, обеспечивающие работу системы с постоянным углом опережения зажигания при отказе основной части схемы электронного блока управления. Скважность импульсов зажигания не зависит от частоты вращения коленчатого вала; ее величина программируется подключением определенных выводов микросхемы к шинам Лог. 0 или Лог. 1 (аналогично программируется величина резервного угла опережения зажигания).

Микросхемы КМ1823ВУ1, КМ1823ВВ1 и КМ1823АГ1 изготавливаются по КМДП технологии с поликремниевыми затворами и выпускаются в 28-выводном металлокерамическом корпусе 2121.28-6.

Основные характеристики микросхем серии КМ1823

	КМ1823ВУ1	КМ1823ВВ1	КМ1823АГ1
Число элементов, шт.	2900	2800	840
Тактовая частота, кГц	4200	520	—

Примечания. Напряжение питания для всех типов микросхем 5±10 % В; потребляемая мощность, не более 0,08 мВт; выходной ток низкого уровня, не более 0,8 мА; выходной ток высокого уровня, не менее 0,4 мА; диапазон рабочих температур —60...+100 °С

Постоянное запоминающее устройство КМ1823РЕ1 выполняется по биполярной технологии с масочным программированием информации и выпускается в металлокерамическом 24-выводном корпусе 2121.24-6. Расположение и назначение выводов микросхемы аналогично БИС ЭППЗУ с ультрафиолетовым стиранием информации КМ573РФ2 и КМ573РФ5, что позволяет использовать ее при отладке схемы электронного блока микропроцессорной системы управления без изменения топологии печатной платы.

Основные характеристики микросхемы КМ1823РЕ1

Информационная емкость	2К×8
Время выборки, нс, не более	100
Напряжение питания, В	5±10 %
Потребляемая мощность, мВт, не более	495
Выходной ток низкого уровня, мА, не менее	18
Диапазон рабочих температур, °С	—60...+100

В настоящее время на базе описываемого комплекта БИС серии КМ1823 разработаны и серийно выпускаются контроллеры микропроцессорных систем зажигания для легковых («Электроника МС2713») и грузовых («Электроника МС2709») автомобилей.

Телефон 77-44-32, Минск

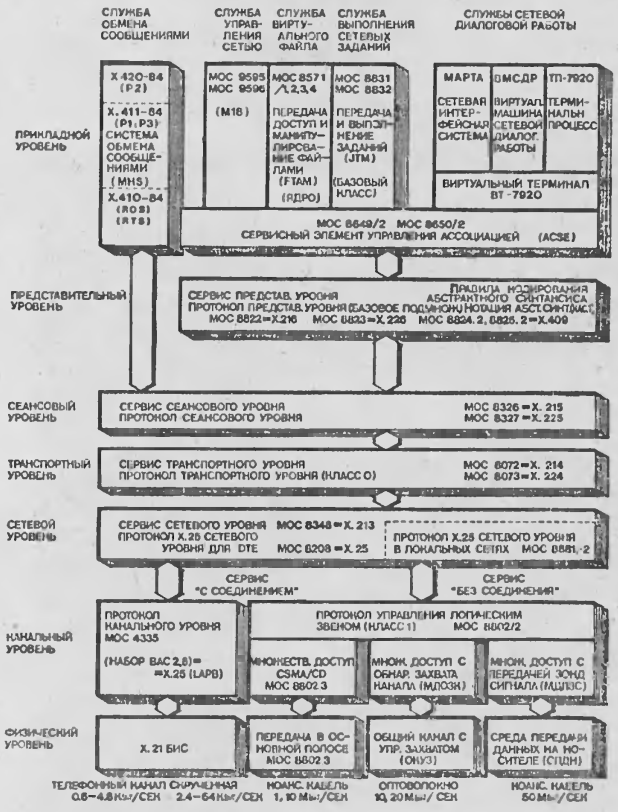
Статья поступила 29.03.88

ИЭВТ АН ЛАТВССР ПРЕДЛАГАЕТ

Если Вы занимаетесь проблемами создания отраслевых сетей, для Вас будет интересна концепция сетевой архитектуры «Интеграл-88» (САИ) (см. рисунок).

Концепция показывает одну из возможностей создания комплекса сетевых средств, позволяющих эффективно решать как специальные задачи, т. е. построение проблемно-ориентированных локальных сетей, так и их интеграции в единую сеть ЭВМ.

СЕТЕВАЯ АРХИТЕКТУРА ИНТЕГРАЛ 88



Основа архитектуры САИ — эталонная модель взаимодействия открытых систем МОС и международные стандарты.

САИ обеспечивает возможность интеграции в единую систему сетей, построенных на базе протоколов MAP/TOP и X. 25.

На основе САИ мы готовы разработать концепцию Вашей отраслевой сети.

Для тех, кого интересует состояние дел в области стандартизации протоколов, ИЭВТ АН ЛатвССР предлагает концепцию протоколов открытых абонентских систем.

Концепция содержит описание и анализ основных существующих документов по сервису, протоколам и интерфейсам всех уровней иерархии. Позволяет оценить состояние дел в области разработки стандартов ВОС, тенденции и проблемы стандартизации.

У нас Вы можете получить консультации по вопросам применения международных стандартов.

Более подробную информацию Вы можете получить по тел. 55-82-46, Рига.

Разработка и передача материалов производится на договорной основе.

УДК 681.3.06

В. А. Крюков

АНАЛИЗ ПРИНЦИПОВ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Догмы спокойного прошлого не годятся для бурного настоящего

А. Линкольн

Введение

Новый термин «объектно-ориентированное программирование» (ООП) приобрел за последнее время достаточную известность в кругу программистов. Этому способствовали публикации в журналах по программированию [1—7]. Однако из семинаров, статей и личных бесед я вынес суждение, что общепринятого мнения по этому вопросу не существует и что в этот термин вкладывают порой самый разный смысл. Относясь с большой симпатией к плюрализму мнений, особенно в науке, тем не менее я нахожу такое положение ненормальным. Именно потребность в осознании единой концепции ООП с общих позиций вызвала необходимость исследования, в ходе которого я стремился добиться наибольшей ясности в этом вопросе.

Забегая вперед, скажу, что эти исследования показали, во-первых, адекватность понятия объект общедисциплинарному понятию система, или модель системы. Во-вторых, методологические принципы ООП оказались очень близкими к концепции Г. Р. Громова об автоформализации профессиональных знаний [7, 8], что несколько облегчило мою задачу провести цепочки рассуждения и обоснования по короткой траектории журнальной публикации, ссылаясь в нужных местах на работы [7, 8].

История и состояние вопроса

Объектно-ориентированное программирование зародилось в ходе научных исследований, проводимых фирмой Xerox в 70-е годы, в основе которых лежала идея активности данных, а не процедур. Эта концепция изначально имела направленность, в первую очередь, на улучшение человеко-машинного интерфейса и повышение уровня разработки прикладных программ. Результатом десятилетнего исследования в фирме Xerox явилось создание языка-системы SmallTalk-80 [1, 9—11] (1980 г. считается годом рождения языка). В 1987 г. создатели языка получили премию ACM за лучшую программную разработку.

Идеология языка SmallTalk-80 связана с концепцией творческой персональной работы пользователя на достаточно мощной вычислительной рабочей станции:

- размер оперативной памяти 0,5 Мбайт (для серьезной работы требуется >1 Мбайт памяти);
- разрядность процессора $\geqslant 16$ бит;
- наличие растрового графического дисплея с высокой разрешающей способностью;
- наличие графического указывающего устройства типа мышь.

В начале 80-х годов промышленность была не готова выпускать аппаратуру, удовлетворяющую этим требованиям, по цене персональной ЭВМ (ПЭВМ). Это сдерживало широкое распространение языка SmallTalk-80. Однако ситуация начала кардинально меняться в середине 80-х годов, когда в результате острой конкурентной борьбы на мировом рынке персональных компьютеров, особенно между фирмами IBM и Apple, характеристики ПЭВМ улучшились до уровня характеристик вычислительных рабочих станций.

В 1984 г. фирма Apple выпустила в продажу 32-разрядный компьютер «Макинтош», на разработку которого оказали прямое влияние исследования фирмы Xerox и в первую очередь ГИО*. Компьютер «Макинтош» отличается от других персональных компьютеров развитым объектно-ориентированным графическим интерфейсом с пользователем, что в совокупности с высокими техническими характеристиками обеспечило ему большую популярность. Фирма Apple реализовала совместимую версию языка SmallTalk-80 на компьютере «Макинтош» и продает его по низкой цене [12].

На персональных компьютерах фирмы IBM долгое время SmallTalk не использовался, пока в 1986 г. фирма Digitalk Inc. не разработала замечательную по экономии памяти версию с названием SmallTalk/V для персональных компьютеров фирмы IBM, модель которых не меньше XT [13].

Наряду с Xerox и Apple исследования, связанные с применением объектно-ориентированного подхода к решению проблем программирования, проводятся во многих фирмах, крупных производителях вычислительной техники и оборудования, в частности: DEC, Hewlet-Packard, Tektronix, ATT, Intel [5, 14]. Его используют также японские специалисты в рамках проекта ЭВМ пятого поколения [15]. С 1986 г. в США и Европе проводятся ежегодные конференции, посвященные ООП, языкам и прикладным системам (в США под эгидой ACM-SIGPLAN) [3, 4].

Чем ООП привлекает к себе все большее внимание и завоевывает все большее число сторонников? Ответ достаточно простой: тем, что основным элементом ООП является объект как модель системы, а понятие системы в XX веке стало центральным инструментом в методологическом арсенале современных прикладных наук.

Кроме того, ООП открывает возможность по-новому решать основные проблемы программирования; улучшение надежности разработки программ, создание механизма управления сложностью программ, повышение производительности программиста, предоставление естественного интерфейса человека с машиной и т. д.

Смена основного элемента программирования неизбежно влечет за собой смену парадигмы программирования, поэтому начнем концептуальное рассмотрение ООП именно с этого вопроса.

Две парадигмы программирования

Отправным пунктом рассуждений является точка зрения, что причина коренных изменений в ООП связана со сменой основной парадигмы программирования.

Согласно «Советскому энциклопедическому словарю» термин парадигма означает исходную концептуальную схему, модель постановки проблем и их решения, методы исследования, господствующие в течение определенного исторического периода в научном сообществе. Здесь это слово применяется в смысле представления о базовых элементах программирования и их взаимосвязях.

Необходимость выделения из всего многообразия представлений программирования определенного минимального набора понятий, составляющих парадигму, связано с тем, что без их усвоения невозможна успешная самостоятельная работа конечного пользователя с программой. Последнее всегда важно, но становится особенно актуальным в условиях широкого распространения ПЭВМ и все боль-

* ГИО — Группа Исследования Образования в научном центре Palo Alto фирмы Xerox.

шего вовлечения специалистов проблемных областей в использование, а также в разработку прикладных систем.

Для начала в качестве диаметрально-противоположного примера рассмотрим основную парадигму процедурно-ориентированного программирования. Эта парадигма возникла на заре становления вычислительной техники, имеет длительную историю развития и в настоящее время является господствующей. Она распространяется на все слои программного обеспечения (ПО): от ОС (управление процессами виртуальной машины) до прикладного уровня. Информатика как наука развивалась именно под ее влиянием.

Парадигма процедурно-ориентированного программирования базируется на представлении об управлении машиной со стороны программы или набора процедур (подразумевается, что машина может быть абстрактной, например Фортран-машина). Характерная взаимосвязь элементов в парадигме показана в виде диаграммы (рис. 1). Выполнение процедур естественно назвать процессом, причем из одного процесса может активизироваться другой, что является главным способом управления общим процессом. В момент активизации может производиться передача данных, которые носят в таком случае название параметров. Окончание выполнения процедуры связано с возвращением результата и функции управления в вызывающую процедуру.

Как создается парадигма? Она формируется на этапе конструирования вычислительной среды, в том числе языков программирования высокого уровня. Основной фактор, учитываемый при конструировании вычислительной среды,— легкость и гибкость отображения предметной области решаемых задач на вычислительные средства. Такая постановка вопроса справедлива, так как между процессом формализации знаний и программированием существует прямая связь: «Процесс записи ранее формализованных знаний и готовой для непосредственного воздействия на машины и механизмы форме получил название «программирование ЭВМ» [7, с. 81].

Применим критерий удобства формализации знаний для сравнения ООП с процедурно-ориентированным программированием, в основной парадигме которого видны следующие недостатки:

1. Парадигма программирования несет в себе пространство процессов («пространство приведений»), а доминирующие представления у людей, лежащие в основе нашего миропонимания, имеют предметный характер. Последнее можно объяснить условиями нашего бытия, но нас в настоящий момент должен интересовать возможный вывод, что несоответствие этих представлений приводит к трудностям решения задач проблемных областей на ЭВМ [7, 16—18].

2. Существует неестественная сложность организации работы пользователя в среде процессов [19]. Например, нельзя из редактора текстов запустить численную задачу или обратиться к базе данных, чтобы результаты поместить в документ. Пользователь должен знать множество режимов работы в разных процессах и пути достижения нужного процесса. Знание таинств перехода из текущего процесса в нужный процессор является непомерной платой для многих пользователей за решение задач их проблемной области. В этой ситуации мало помогают так называемые «дружественные» средства общения с ЭВМ, которые помещены в недружественную среду.

3. Парадигма сильно снижает основную способность данных моделировать реальный мир. Даже если данные являются глобальными, а не передаются в качестве параметров, они все равно играют в этой парадигме роль параметров процедур или функций. Данные «растворены» в программе и формулировки типа

АЛГОРИТМЫ + СТРУКТУРЫ ДАННЫХ = = ПРОГРАММЫ

не восстанавливают статус-кво.

4. Невозможно без привлечения дополнительных представлений создать механизм сохранения между сеансами интерактивной работы контекста задачи, представляющего собой значение всех данных, которые как бы «умирают» после окончания выполнения процедуры. Привлечение дополнительных представлений о файловой системе или базе данных влечет за собой дополнительную сложность работы с ЭВМ.

5. Феномен времени не находит своего выражения в данной парадигме. Изменение времени в нашем сознании связано с изменением состояний каких-либо объектов, например природы весной, или одного объекта, например показания Ваших наручных часов. Начало или окончание процесса выполнения процедуры не может служить моментом отсчета, поэтому без дополнительных представлений нельзя завести механизм с функцией отсчета времени. Феномен времени важен для имитационных задач и для построения вычислительных систем реального времени и ОС.

Необходимо сделать небольшое отступление по поводу известного факта роста стоимости программного обеспечения и повсеместного срыва сроков выполнения заданий на его разработку [16—18]. По мнению многих, причина этого кроется в несоответствии реализаций программ спецификациям на них, а спецификаций — представлениям пользователей, что влечет за собой постоянное внесение исправлений. Эту проблему можно назвать субъективной, потому что она связана с теми или иными субъективными представлениями. Однако практика показывает, что существует еще и объективная причина — постоянное развитие и изменяемость наших знаний о предмете формализации. Необходимость постоянного отслеживания наших знаний по конкретному вопросу и приводит к потребности модификации ПО. Поэтому программы, особенно прикладного характера, с момента своего создания и начала эксплуатации объективно требуют вечного сопровождения. Эффективно реализовать сопровождение в рамках процедурно-ориентированного программирования не удается в силу вышеназванных причин.

Отсюда можно сделать выводы, что более подходящим инструментом формализации знаний на ЭВМ была бы вычислительная среда, основанная на другой парадигме, базирующейся на концепции, которая приближена к понятию системы, или модели системы, получившей общедисциплинарное распространение.

Такая постановка вопроса приводит к парадигме ООП, в основе которой лежит идея активности данных, а не процедур, в чем можно усмотреть обращенность новой парадигмы по отношению к вышерассмотренной. Диаграмма новой парадигмы изображена на рис. 2. На первый

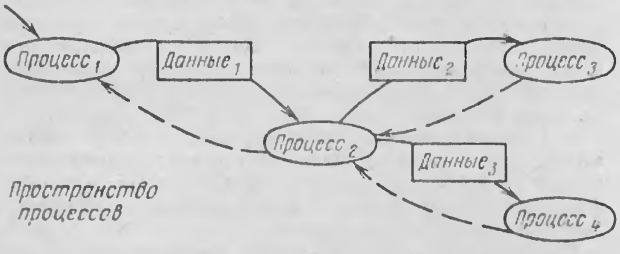


Рис. 1.

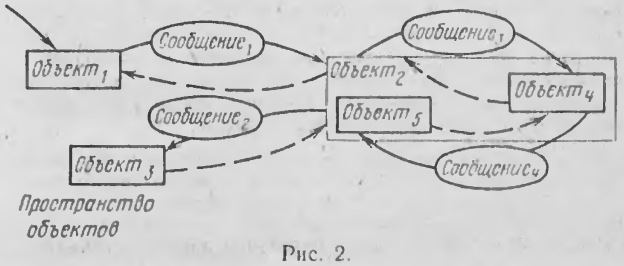


Рис. 2.

взгляд, рис и 2 выглядят идентично, что может послужить основанием для подозрения, что две парадигмы мало отличаются друг от друга. Однако это не так. Достаточно сказать, что объекты во время исполнения программы могут менять свое значение, и это отражает их сущность как данных. Для процесса внутри парадигмы процедурно-ориентированного программирования самомодификация кода является плохим стилем программирования и применяется крайне редко.

Новая парадигма включает в себя три понятия: объект, сообщение и метод, определения которых извлечены из работы [9] и помещены в приложении. К сожалению, эти определения не несут в себе большой семантической нагрузки, и поэтому целесообразно рассмотреть эти понятия с самых общих позиций.

Ключевым понятием является абстракция под названием о б ъ е к т, которая призвана отображать элемент проблемной области в элемент вычислительной среды. Поэтому рассматривать объект можно в двух аспектах: во-первых, как элемент вычислительной системы, а во-вторых, как общее понятие системы разных проблемных областей.

Сначала рассмотрим второй аспект. Воспользуемся для этого положениями о системе такой науки, как системология, утверждающей, что система, а в нашей терминологии объект, определяется своей структурой и поведением [20]. Утверждается также, что понятия структуры и поведения — столь же фундаментальные и неопределяемые понятия, как «пространство» и «время» в физике или «множество» в математике. Понятие структура носит относительный характер, так как структура объекта может считаться состоящей или из неделимых элементов атомарного уровня, или из сложных объектов, в свою очередь, состоящих из простых и менее сложных объектов. Например, **Объект**$_2$ (см. рис. 2) состоит из двух объектов: **Объект**$_4$ и **Объект**$_5$.

Вернемся к первому аспекту рассмотрения объекта. Понятие объект можно определить формально через кортеж с помощью следующих правил:

⟨объект⟩::=(⟨список компонентов⟩, ⟨модель поведения⟩). (1)

⟨список компонентов⟩::=[⟨компонент⟩]. (2)

⟨компонент⟩::=⟨объект⟩:⟨атомарный объект⟩. (3)

⟨атомарный объект⟩::=(⟨значение⟩, ⟨априорное поведение⟩). (4)

Формальный подход дает совпадение определения объекта с основными положениями о системе из системологии: рекурсивность определения прямо связана с относительностью понятия структура, а невозможность формального доопределения априорного поведения атомарных объектов указывает на их фундаментальность. Под априорным поведением понимается набор операций, внутренне присущих атомарному объекту.

Интересно отметить роль формул (1) и (4). Формула (4) представляет объект с внешней стороны, а формула (1) раскрывает внутреннее устройство объекта. Под моделью поведения понимается последовательность сообщений, которую посылает объект, чтобы выполнить какую-либо деятельность, о чем речь пойдет ниже.

Вернемся к рассмотрению устройства новой парадигмы, о котором можно сделать ряд заключений общего характера. Например, если объект получает имя в вычислительной системе, то он становится глобальным и к нему можно обращаться по его уникальному имени. Поэтому можно считать, что все глобальные объекты равнодоступны, будь то системные элементы, внешние устройства, графические формы, тексты или объекты формализуемой области знаний. Неглобальные объекты могут существовать только в случае вхождения в состав глобальных, иначе они по сути своей представляют «мусор».

Объект имеет возможность взаимодействия с внешними по отношению к нему объектами. В качестве такого средства взаимодействия в ООП используется механизм посылки и приема сообщений, который, по-видимому, плохо соответствует представлению о взаимодействии систем, но лучшего ничего не придумали. Читатель может предложить свой механизм взаимодействия.

Итак, для того чтобы заставить объект произвести какие-либо действия, т. е. начать выполнение программы, и получить от него результат, необходимо этому объекту послать сообщение-запрос. В ответ на запрос объект сам может послать последовательность сообщений другим объектам. В таком случае цепочка сообщений удлиняется, а трасса сообщений может ветвиться, как на рис. 2. Таким образом, организуется совместная деятельность объектов внутри вычислительной системы. Последовательность сообщений, которые будут посланы одним объектом ряду других объектов, а может быть и самому себе (рекурсия), как реакция на получение сообщения и является моделью поведения объекта. Реакции объекта могут быть различными в зависимости от вида полученного сообщения. В данном случае к (1)—(4) добавится уравнение

⟨модель поведения⟩::=

⟨набор последовательностей сообщений⟩. (5)

Общая характеристика новой парадигмы:

парадигма несет в себе естественную картину пространства объектов, взаимодействие которых может быть организовано по причинно-следственным связям;

так как в любой момент работы в объектно-ориентированной среде пользователю доступны все глобальные объекты, то он может легко переходить от одного вида деятельности к другому: редактированию текста, просмотру и анализу данных, выполнению расчетных работ, обработке изображений и т. д. Переход производится однотипно;

в противоположность традиционной парадигме данные локализованы и структурированы естественно;

сохранение контекста задачи между сеансами интерактивной работы получает естественное выражение: объект — не процесс и исчезнуть не может;

Для отсчета времени может быть заведен специальный объект — таймер, который может или посылать сообщения, или отвечать на запросы о времени.

Еще один положительный момент: в рамках новой парадигмы, вероятно, можно отказаться от представления о расщеплении памяти на первичную и вторичную и ограничиться представлением о пространстве объектов (см. рис. 2), в которых находится вся информация. Об иерархии памяти будут знать только архитекторы и производители вычислительных систем.

Проведенный анализ показывает, что парадигма ООП имеет преимущества перед парадигмой традиционного процедурного программирования при формализации знаний, что объясняется сокращением так называемого семантического разрыва. В данном случае речь идет о сокращении разрыва между принципами моделирования реальных объектов и принципами, лежащими в основе языков программирования. Этот разрыв можно назвать вторым семантическим разрывом, так как Г. Майерс [21] уже ввел понятие семантического разрыва между языками программирования и архитектурой ЭВМ. Заметьте, что второй семантический разрыв граничит с первым.

Для сокращения первого семантического разрыва Г. Майерсом показана необходимость наличия в программировании понятия объекта (!). Есть и другие параллели: требование одноуровневой памяти, дескрипторы данных, показывающие природу объекта. Такое совпадение в смежных областях информатики не может быть случайным.

Надо сказать, что не всегда смена парадигмы приводит к положительному результату. Например, в парадигме Пролог процедурный механизм заменен на механизм логического вывода, которого оказалось недостаточно для изначальной задачи языка — моделирования знаний [22]. Почти о том же говорится в работе [7, с. 87]:

«...с расширением областей приложений ЭВМ... традиционно неразрывная со времен «программы математизации знаний Пифагора из Самоса» концептуальная связь двух фундаментальных понятий: **формализация и логическая доказательность** — не сохраняется».

Процедурность — специфическая часть наших знаний о мире, что согласуется с концепцией об особом назначе-

нии ПО формализации знаний. Поэтому, чтобы ПО было способно вбирать в себя знания, процедурность должна быть неотъемлемой частью языков программирования.

Но процедурность в чистом виде была представлена на рис. 1, поэтому создается впечатление, что предложение новой парадигмы ООП автоматически исключает процедурность. Нет, этого не произошло. По этому поводу говорят, что процедуры инкапсулированы в объекте. Согласно формулам (1)—(5), процедурность локализована внутри объекта и задается последовательностью посылок сообщений. Однако справедливости ради необходимо отметить, что в языке SmallTalk-80 процедуры, или, следуя терминологии самого языка, методы, локализованы не в объектах, а в классах, которые его описывают. Тогда (5) преобразуется в формулу

⟨модель поведения⟩::=методы классов по наследованию.

Далее рассматриваются понятие наследования, особенности языка программирования, системы программирования и методов программирования в рамках новой парадигмы на базе языка-системы SmallTalk-80 по причине, указанной во введении.

Характеристики объектно-ориентированных языков

Принято указывать на четыре свойства объектно-ориентированных языков [23, 24]: упрятывание данных (инкапсуляция), абстракция данных, механизм наследования, динамическая привязка.

Упрятывание данных используется для конструирования объектов. Данные, составляющие объект, становятся доступными внешним процедурам только через интерфейсный аппарат сообщений. Поэтому они могут быть изменены и (или) переданы вовне только через обращение к этому интерфейсу объекта. Внутренняя информация объекта как бы прячется от «внешнего мира». В объектно-ориентированном языке интерфейс объекта определяется набором селекторов множества возможных сообщений, фактически таблицей имен внутренних процедур. Обычно для каждого вида, или, вернее сказать, класса, объектов устанавливается один набор селекторов. Например, в SmallTalk-80 объекты класса **Point** (на понятии класса остановимся более подробно в следующих пунктах) имеют две переменные **X** и **Y** и понимают сообщение

x:50,

где **x:** — селектор сообщения, а 50 — его аргумент. Посылка такого сообщения объекту класса **Point** вызовет изменение соответствующей переменной внутри объекта. Другого способа на языковом уровне изменения внутренних переменных не дано. Шаблоны сообщений пользователь может узнать или восстановить в памяти во время интерактивной работы с помощью просматривателя классов.

Абстракция данных тесно связана с инкапсуляцией и поддерживает механизм абстрактных типов данных, состоящих из описания структуры объекта и набора процедур, принадлежащих объектам только этого абстрактного типа*. Понятию абстрактного типа данных в языке SmallTalk-80 соответствует понятие класса, только процедуры именуются словом методы. Этой терминологии мы и будем придерживаться дальше по тексту.

Таким образом, все объекты в языке SmallTalk-80 принадлежат какому-либо классу и являются экземплярами соответствующих классов. В таком случае сущность программирования заключается в создании и описании классов (или подклассов). В языке SmallTalk-80 программисту предоставляется большое разнообразие уже готовых классов, малую часть которых составляют следующие:

ArrayedCollection	**DisplayObject**
Association	**Font**
Boolean	**Character**
Directory	**Float**
Fraction	**Point**
Integer	**Rectangle**
Menu	**Stream**
Pen	**String**

Назначение этих классов видно из их названия, например класс **ArrayedCollection** представляет множество объектов, аналогичных типу массив в других языках. Класс **String** представляет множество строк символов, но они также принадлежат классу **ArrayedCollection**, так как класс **String** является подклассом класса **ArrayedCollection**, а класс **ArrayedCollection**, в свою очередь,— суперкласс класса **String.**

Совокупность однонаправленных связей вида класс — суперкласс между классами языка SmallTalk-80 образует иерархическую систему. В корне иерархии находится класс **Object**, фиксируя тот факт, что в системе все считается объектами. Таким образом, понятие объект трактуется очень широко, и любой класс также является объектом и представлен как экземпляр класса классов (метакласса). Классу как объекту можно послать сообщение, например, чтобы создать экземпляр класса.

Для пользователя древовидная структура классов может быть представлена как механизм, подобный классификации объектов по иерархическому принципу, с которым он хорошо знаком из научных дисциплин по естествознанию. По сути иерархическая структура классов — независимая дополнительная (ортогональной) парадигма, пересекающаяся с основной парадигмой на уровне экземпляров классов. В объектно-ориентированных языках по этим ортогональным плоскостям производится деление технологических зон пользователя и программиста: пользователю — содержательные объекты, программисту — классы. Можно сказать, что если содержательные объекты являются объектами для пользователя, то классы — объектами для программиста.

Рассмотренные иерархические связи между классами используются в механизме наследования.

Механизм наследования позволяет объекту использовать описание не только непосредственного класса, экземпляром которого он является, но также суперкласса и всех остальных классов, расположенных сверху по цепочке иерархии. Например, класс **String** уточняет информацию о строках как о массивах, содержащих литеры (экземпляры класса **Character**), а из класса **ArrayedCollection** наследуются методы доступа к символам строк как к элементам массива.

Большое разнообразие готовых классов и возможность объекта наследовать свойства позволяют программисту разрабатывать программу, создавая подклассы, и тем самым уточнять описание объекта как программной модели. Такой режим работы был бы значительно затруднен, если бы в языке SmallTalk-80 не было динамической привязки.

Динамическая привязка означает, что адрес необходимого метода формируется во время исполнения, а не компиляции. В языке SmallTalk-80 метод ищется по цепочке суперклассов объекта в момент посылки сообщения. Адреса глобальных объектов тоже ищутся во время исполнения. Поэтому язык SmallTalk-80 относится к интерпретируемым языкам, хотя в нем существует компиляция до промежуточного кода. В этом коде методы хранятся в классах.

Описанный механизм поиска метода, соответствующего селектору сообщения, позволяет утверждать, что язык SmallTalk-80 — язык с типизацией данных. Это весьма важно для поддержания правильности использования данных и, как следствие, для более высокой надежности ПО. Однако есть некоторое отличие от других языков со строгой типизацией данных, например от Паскаля,— контроль над типами данных, который обычно проводится транслятором во время компиляции, в языке SmallTalk-80 заменен на управление типами данных во время исполнения программы.

Динамическая привязка ведет к снижению скорости выполнения. Но проигрывая в скорости компилируемым языкам, язык SmallTalk-80 выигрывает в улучшении тех-

---

* Существует также более широкое понимание абстрактных типов данных, расширенных за счет раздела спецификаций. Такая абстракция данных используется в теоретических работах для доказательства правильности программ [25].

нологии программирования. На технологию программирования влияет множество факторов, но относительно динамической привязки можно сказать, что она влияет положительно, так как внесение локальных изменений не вызывает повсеместную волну изменений и не требует перекомпиляции всей программы. Это означает, что ПО обладает свойством пластичности для быстрого внесения изменений во время анализа программистом результата предыдущего шага в творческом процессе. Если в программу внесена ошибка, то благодаря свойству инкапсуляции она локальна и ее легче обнаружить. Впрочем, можно заметить, что эти свойства, присущие языку, являются следствием объектно-ориентированной парадигмы.

Разработчики языка SmallTalk-80 при проектировании стремились получить оптимальные характеристики совокупности средств программирования, а не отдельных ее компонентов. В качестве критерия оптимальности выбиралась гибкость и простота работы в SmallTalk-80 как в системе, о чем речь пойдет в следующих разделах. Но сначала рассмотрим вопрос, насколько объектно-ориентированный стиль доступен в других языках.

Эволюция языков в направлении ООП

Логично предположить, что уже давно многие программисты стихийно или сознательно придерживаются объектно-ориентированного стиля, применяя те встроенные механизмы языка, которые наиболее для этого подходят. Например, в Паскале для моделирования объекта можно использовать механизм записей (Record), а инкапсуляцию процедур, классы и наследование свойств можно имитировать вручную с помощью организационных мер и определенной доли самодисциплины. Однако поступать так сегодня равносильно имитировать рекурсивный вызов процедур в языке Фортран, имея встроенный механизм рекурсии в языке более высокого уровня, скажем, в том же Паскале.

Встраивание в традиционные языки механизмов описания классов, наследования, передачи сообщения, создания объектов в работе [23] названо эволюцией языков в объектно-ориентированном направлении. Наибольший интерес представляет эволюция языка Си — распространенного языка системного программирования.

Объектно-ориентированный программный продукт C++ фирмы AT&T [26] вызывает к себе интерес в первую очередь тем, что фирма AT&T — прародительница языка Си. Существует также продукт фирмы Productivity Products International под названием Objective-C [12, 23]. Он состоит из препроцессора, обрабатывающего тексты на самом Objective-C в тексты на обычном Си, и библиотеки 25 классов, написанных машинно-независимым образом. Новые классы описываются в специальных файлах — определителях класса. В выражениях языка Objective-C могут совмещаться операторы обычного языка Си (арифметические операторы, вызовы функций) и операторы посылки сообщения.

Оба продукта получают в настоящее время быстрое и широкое распространение, что подтверждает целесообразность включения концепций ООП в традиционные языки. Это направление обеспечивает языковую преемственность для большой группы программистов и сохранность производительности компилируемых языков.

Следующий раздел представляет собой маленький экскурс в SmallTalk, что позволит читателю ощутить «привкус» языка. Для получения исчерпывающей информации необходимо обратиться к работам [9, 10].

Немного о SmallTalk

Приведем два конкретных примера из языка SmallTalk. Математическое выражение sin (3*beta) будет записано так:

3*beta sin

которое трактуется как посылка двух сообщений: «* beta» и «sin». Первое сообщение посылается числу 3 и состоит из селектора «*», по которому в классе **SmallInteger** отыскивается соответствующий метод, и аргумента «beta». Произведению 3×beta посылается сообщение «sin». Значение выражения может быть сохранено с помощью обычного присваивания*:

x:=3*beta sin.

Выражение должно оканчиваться точкой.

При первом знакомстве такой способ записи вызывает трудности прочтения из-за смеси инфиксной и постфиксной нотаций, однако им быстро овладеваешь, когда приучаешься думать в терминах посылки сообщения объекту, стоящему слева. Нематематические выражения имеют более естественный вид, что попытаемся продемонстрировать ниже.

Существует стандартное сообщение new, посылаемое классу для создания его экземпляра. Выражение

Turtle:=Pen new.

создает экземпляр класса Pen, который имитирует «черепашку» языка Лого, и присваивает ему имя Turtle. Теперь, имея в системе объект с именем Turtle, можно посылать ему сообщения. Выражение

```
Turtle home;
north;
black;
go:60.
```

— каскад из четырех сообщений, разделенных «;». В результате выполнения запросов сообщений черепашка сначала займет исходное положение с ориентацией на верх экрана, а затем переместится на 60 единиц растра, оставляя за собой черный след.

```
|string count countLine sortColl :   "локальные переменные"
 count     := 0.                      "счетчик объектов"
 countLine := 0.                      "счетчик строк"
 sortColl  := SmallTalk keys  asSortedCollection.
              "извлечение всех ключей словаря SmallTalk
               и сохранение их как отсортированный массив"
 sortColl do: [ :each :       "цикл  по  всем  элементам массива"
    count     := count + 1.
    countLine := countLine + 1.
    string    := each asString.
    countLine  = 51 ifTrue: [       "Условие конца страницы"
         countLine := 0.
         Terminal  bell.
         (Prompter prompt:'continue?'
                   default:'yes') = 'no' ifTrue:[^'Good by']
                                "опрос на продолжение печати"
         Printer
           nextPutAll: '     ';    "вывести пустую строку"
           nextPutAll: count printString, '     ';
                                "вывести порядковый номер"
           nextPutAll: string;
           next: 30 - string size put: Space;
                                "выравнить правый столбец"
           nextPutAll:((SmallTalk at: each) class printString);
                                "вывести название класса"
           cr ].                "перевести строку"
```

Рис. 3.

* Символом присваивания в языке SmallTalk-80 является символ стрелка влево, который отсутствует на стандартной клавиатуре. Поэтому в реализации этого языка SmallTalk/V фирмой DigitTalk используется более традиционный символ присваивания «:=». Этому примеру последуем и мы.

Далее, после первого знакомства с основами синтаксиса языка SmallTalk в качестве примера можно привести текст небольшой программы, которая распечатывает на принтере имена глобальных объектов в системе и поэтому может оказаться полезной, когда Вы встретитесь с неизвестным пакетом дискет со SmallTalk. Имена глобальных объектов в программе помещены в словарь с названием SmallTalk. Распечатка (рис. 3) ведется в два столбца (слева — имя объекта, справа — имя соответствующего класса).

Программа набирается в рабочем окне и непосредственно выполняется интерпретатором, инициализация которого производится командой **«doit»** из меню. Приведенный текст программы после небольших изменений может быть включен в состав методов экземпляров (словарь SmallTalk) класса Directory. Селектором сообщения может быть, например, слово printKeys, тогда распечатку объектов в системе получают с помощью набора и инициализации одной строчки: SmallTalk printKeys.

SmallTalk-система

SmallTalk-80 — это не только язык, но и большая* система [9], включающая в себя объекты, которые являются в обычных вычислительных системах компонентами ОС: диспетчер памяти, монитор процессов, компилятор, интерпретатор, отладчик, файловая система** и т. д.

Система SmallTalk-80 строится по принципу наращиваемого ядра. В ядро входят диспетчер памяти, интерпретатор и набор встроенных методов, обеспечивающих базовые функции: ввод-вывод, арифметика, графические примитивы и т. д. В ядре также формируются базовые классы. Абстрактная машина ядра — основа для реализации оставшихся компонентов системы: компилятора, отладчика, редакторов (текстового и графического), декомпилятора и файловой системы. Все перечисленные компоненты пишутся на самом SmallTalk и формируют оболочку системы, состоящую из набора классов более высокого уровня, чем классы ядра. Оболочка является достаточно «раздутой», чтобы максимально приблизить их уровень абстракции к уровню формализуемых проблемных областей и минимизировать работу программиста с помощью готовых программных шаблонов.

Проблемное ПО создается или синтезом новых классов, или определением подклассов, которые специализируют, если хотите, адаптируют ПО на проблемную область. Причем, могут использоваться сообщения к уже существующим системным объектам, как, например, в рассмотренной программе сообщение к объекту Prompter. Таким образом, хотя системные и прикладные классы находятся на разных уровнях абстракции, между ними нет принципиальной разницы: реализуются они на основе единого механизма объектов, сообщений, классов и наследования. Новые классы становятся частью системы наравне с системными классами.

Система SmallTalk-80 содержит помимо получивших распространение и ставших штатными интерфейсных средств (мышка, окна) ряд специфичных, помогающих одновременно и ориентироваться в системе, и модифицировать данные, и создавать программы. Поэтому назовем их инструментально-интерфейсными средствами. В системе их два: просматриватель классов и инспектор данных.

Просматриватель классов — системное средство доступа и отображения на экран иерархии классов и содержимого самих классов (описания структуры экземпляров класса и методов класса), а также средство изменения классов. Просматриватель может инициализироваться из системного меню, но удобно держать его постоянно в одном из окон на экране дисплея. Окно (window) просматривателя может иметь (в зависимости от реализации) 3...5 частей, которые получили название рамок (pane). Содержимое рамок окна логически взаимосвязанно. В одной из рамок помещается список имеющихся классов.

Выбор с помощью мышки одного из классов приводит к автоматическому показу в соседней рамке справа списка селекторов методов этого класса. Если во второй рамке также произвести выбор, то в нижней рамке появится текст соответствующего метода. Текст метода можно редактировать штатными средствами. Создание нового класса или нового метода производится с помощью «выталкиваемого» меню из соответствующей рамки.

Инспектор данных — системное средство изобразительного представления и изменения содержимого объекта. Инспектор создается путем посылки сообщения inspect интересуемому объекту. В результате на экране дисплея появляется окно, содержащее название внутренних переменных и их значения. Значения могут быть изменены с помощью штатных средств редактирования.

Имеющиеся интерфейсные средства в SmallTalk-80 построены таким образом, чтобы в наибольшей степени удовлетворить принципу — каждый объект можно представить осмысленно для наблюдения и манипулирования. Этот принцип стал общепризнанным, и исследования в этой области еще не закончены, но по всей видимости в SmallTalk-80 впервые была сделана попытка создания языка изображений для объектов. Это потребовало развития самых разнообразных и совершенных средств работы с интерактивной графикой, которые составили графическое ядро языка SmallTalk-80.

Графическое ядро построено на основе модели черно-белой растровой графики. Это позволяет обрабатывать и сочетать на одном изображении графические объекты различной природы: тексты, чертежную графику, картинки, математические графики.

Растровые изображения с прямоугольными границами описываются экземплярами класса **Form** (форма). Базисные операции над формами представлены в классе **BitBlt**, что является отходом в некотором смысле от рассмотренной объектно-ориентированной парадигмы: процедура содержит объект. По словам авторов, это сделано так, поскольку операции **BitBlt** комплексны. Все текстовые и графические объекты на экране создаются и модифицируются с помощью операций **BitBlt**: копирования фрагментов, отсечения по прямоугольной границе, наложения полутона и отображения на фон по определенным правилам. Класс **BitBlt** в некотором смысле подобен механизму преобразования (transformation) в международном стандарте на графическое программное ядро GKS [27].

Для удобства представления форм и задания операций над ними введены классы **Point** (точка) и **Rectangle** (прямоугольник), которые могут также использоваться для объектов прикладной области.

На базе перечисленных классов описываются классы разнообразных графических объектов самого широкого диапазона. Наибольший интерес из них представляет **DisplayText** — изображение текстов; **Line** — отрезки прямых, заданных концевыми точками; **LinerFit** — ломаные линии; **Arc** — четверти окружности; **Spline** — кубические сплайны. В принципе можно сконструировать класс графических объектов любой сложности.

Необходимо отметить удобный режим работы в системе, который близок к естественному режиму работы за письменным столом. Можно, повинуясь ходу мыслей, прервать один вид работ, например редактирование текста, обратиться к другому окну, например за справочной информацией, а затем вернуться к прежнему окну, продолжая прерванную работу.

После краткого описания системы SmallTalk-80 можно сделать вывод, что язык SmallTalk-80 имеет очень сильную поддержку в виде обширного набора классов, доступных пользователю.

Язык и система программирования составляют среду программирования, качество которой непосредственно влияет на методологию программирования.

---

* В статье [23] дается оценка объема SmallTalk-80 в 4 тыс. строк

** SmallTalk обычно не поддерживает своей файловой системы, а обеспечивает интерфейс к файлам основной ОС машины.

Методология программирования в объектно-ориентированной среде

Методология программирования в объектно-ориентированной среде — это недостаточно изученная область, потому что ООП существует всего несколько лет.

Поэтому интерес представляет точка зрения по вопросу методологии самих авторов языка SmallTalk-80. Наиболее определенно эту точку зрения изложила Адель Гольдберг [28]:

«Разрабатываемые с исследовательской целью среды программирования специально проектируются с расчетом на отдельного программиста, который намеревается создать большую, сложную прикладную программу, не пользуясь преимуществами полностью определенного точного проекта. Такой программист должен предусматривать быстрое обращение между написанием кода, тестированием и затем модификацией кода, чтобы проектирование могло развиваться итерационно. Этот программист создает прототип реализации (prototypical implementation). Если результаты этой работы могут «читаться» как описательная документация, тогда программист может использовать прототип для того, чтобы точно определить необходимые условия и спецификации реализации. Это означает, что удачная исследовательская среда программирования должна обеспечить средства, которые поддерживают написание программы так, чтобы как фаза выполнения, так и фаза реализации программ были доступны пользователям, которым они предназначены».

Достаточно ясна концепция технологии программирования, в которой функциональные роли распределены следующим образом:

1) программист последовательно и целенаправленно уточняет прототип от замысла (спецификации на конечный продукт) до конкретной реализации, все время оставаясь в рамках языка программирования;

2) пользователь-заказчик, которому доступно «все», активно участвует в разработке программы и контролирует ее качество непосредственно в фазе создания.

Статья А. Гольдберг является проблемной, и в ней нельзя найти ответ на все вопросы о методологии ООП. Тем не менее, нарисованная картина в большой степени совпадает с технологией автоформализации знаний [7, 8]. Отличие только в расстановке акцентов на психологической роли пользователя в процессе программирования: если для А. Гольдберг пользователь скорее заказчик, то по Г. Р. Громову он еще и конструктор ПО. Такое отличие ролей, отводимых пользователю, я бы объяснил не только спецификой американской и советской практики разработки программного продукта, но и тем, что создатели языка SmallTalk не достигли первоначально намеченного уровня его доступности для пользователя, когда он способен самостоятельно формализовывать свои знания. Поэтому необходимо разобраться, с какими трудностями пришлось встретиться создателям SmallTalk на пути к этой цели.

Первая трудность, которая возникает перед разработчиками программного обеспечения в ООП, состоит в стихийном переносе методов процедурно-ориентированного программирования в ООП по причине консерватизма и инерции мышления самих программистов. Мышление категориями парадигмы процедурно-ориентированного программирования в рамках нового подхода вполне возможно, в чем мы убедились на примере класса BitBlt в графике. Конечно, самые общие принципы сохраняют свою значимость и при новом подходе, например принцип проектирования сверху вниз, который применим в любой эвристической деятельности, каковой является и программирование. Окончательных рекомендаций по методологии в объектно-ориентированной среде следует ожидать только после всесторонних исследований и широкой прикладной практики.

Итак, предлагается рассмотреть два аспекта проблемы обеспечения деятельности: I — пользователя; II — программиста.

I. Понятие пользователя в нашем рассмотрении распространяется на группу людей инженерной профессии, которые применяют ЭВМ регулярно в своей профессиональной деятельности, в то же время сам характер такой деятельности является творческим (эвристическим). Быстрая и эффективная разработка ПО для такой группы пользователей сейчас наиболее актуальна, а создание инструментальных средств для этого представляет научную проблему [8].

Пользователя обозначенной группы при решении проблемной задачи интересуют два вопроса, сформулированных в самом общем виде:

1. Каков состав и структура моделируемого объекта (установки, явления)?

2. Удовлетворяет ли объект функциональным спецификациям?

Второй вопрос лучше переформулировать так: Как ведет себя модель в тех или иных условиях? Для получения ответа на этот вопрос проводятся натурные испытания, что слишком дорого или невозможно, или модельные испытания на ЭВМ. По результатам испытания на ЭВМ делается заключение с определенной степенью достоверности об удовлетворении требованиям функциональной спецификации.

Оба вопроса пользователя взаимосвязаны, но их необходимо методологически различать. Тогда для решения обозначенных вопросов программа логически делится на две части, а для разработки программ и работы с программой требуется наличие двух противоположных по характеру инструментально-интерфейсных средств. Тенденция разделения ПО на два вида наметилась еще в традиционном процедурно-ориентированном программировании. Это базы данных и пакеты прикладных программ, а средства для их разработки — СУБД и инструментальные средства программирования соответственно. Существование подобных специализированных средств можно обнаружить также и в SmallTalk — это инспектор и просматриватель класса.

Первые программные средства отвечают, во-первых, за синтез структуры объекта в автоматизированном режиме и, во-вторых, за представление его для наблюдения на экране дисплея в осмысленном виде. Назовем такие программные средства конструктором. Отметим, что инспектор в языке SmallTalk-80 способен выполнять вторую функцию конструктора полностью, а первую — частично, изменяя значения переменных.

Можно показать первичность понятия объект по отношению к понятиям класс в процессе работы пользователя. Это продиктовано тем, что пользователь работает в плохо структурированной области и классификацию объектов этой области может навести только по результатам профессиональной деятельности. Тогда с точки зрения логики и улучшения технологии целесообразнее было бы создание класса проводить уже после синтеза объекта, а не во время программирования, когда программист наперед описывает объект со слов пользователя, создавая класс.

Возникает противоречивое положение, что должно быть вначале: класс или объект как его экземпляр. Речь идет об обеспечении свободы творчества пользователя, независимого от ограничений, связанных с программой. Пожалуй, это главная проблема в программировании в целом, но в ООП ее необходимо решить, если до конца следовать объектно-ориентированной идеологии. Возникающие трудности при этом главным образом связаны не с механизмом создания класса постфактумом, а с описанием модели поведения объекта: классы хранят методы, совокупность которых и является моделью поведения объектов.

Написание методов внутри классов было и остается до сих пор прерогативой программиста. Доверить пользователю прямую работу не только со структурой объекта, но и с методами не представляется реально возможным. Здесь видятся две взаимосвязанные проблемы. Во-первых, структура объекта и значения его составляющих несут

в себе информацию статической природы, а в методах заключается информация о динамике процессов, связанных с поведением объектов. Поэтому формализация знаний о поведении объектов на порядок труднее, чем формализация их структуры. Во-вторых, как уже отмечалось, механизм обмена сообщениями, который лежит в основе построения методов, не адекватен способам описания взаимодействия систем проблемных областей. Возможно, это несоответствие имеет фундаментальный характер по причине дискретности и последовательности вычислений в ЭВМ, тогда как в природе все непрерывно и протекают параллельные процессы.

Естественно искать решения этой проблемы также в двух направлениях:

1) обеспечение компьютерной грамотности если не в средней, то хотя бы в высшей школе, где и готовятся специалисты проблемных областей;

2) создание по конкретным отраслям знаний инструментально-интерфейсных средств работы с процедурной частью программ (например, по механике, физике, биологии, экономике), обеспечивающих интерфейс пользователя на уровне его профессиональных представлений.

Решение этих задач означало бы решение многих практических проблем в области программирования и в первую очередь проблемы острой нехватки программистов.

Перейдем к рассмотрению вопроса обеспечения деятельности программиста в объектно-ориентированной среде.

II. Ранее мы уже коснулись вопроса средств создания программ, но кто бы ни разрабатывал программы (сам ли пользователь или программист), не менее актуальным остается вопрос способов создания программы.

В работе [29] выделяются три формы программирования: синтезирующее, сборочное, конкретизирующее. Остановимся сначала на двух последних формах программирования, которые реализуют принцип многократного использования программного продукта. Эти формы в ОПП имеют много общего. Поэтому им можно дать название кооперативного программирования, так как они кооперируют усилия многих программистов. Поясним это на примере. Рассмотрим типичный случай разработки одной комплексной программы несколькими программистами. Каждый программист на языке SmallTalk реализует специализированную программную часть на своей персональной ЭВМ. В момент объединения программных частей в одну программу требуется осуществить передачу объекта с машины на машину.

Имеется в виду, что объединение частей будет производиться не в рамках распределенной локальной сети, когда от объекта к объекту может быть наложена межмашинная передача сообщений, а на отдельно стоящей автономной (stand-alone) ЭВМ.

Возникает вопрос, в каком виде должен быть передан объект. Если это экземпляр класса, тогда процесс его включения в программу будет подобен сборочному программированию. Необходимо отметить, что объект должен передаваться со своим классом.

Если передавать только класс, то создание экземпляра этого класса можно сравнивать с адаптацией многопараметрической универсальной программы к особым условиям применения в процедурно-ориентированном программировании, что аналогично конкретизирующему программированию.

Но на пути создания условий кооперативной работы программистов даже в однотипных объектно-ориентированных средах, но на разных вычислительных установках стоят большие трудности, связанные с переносом объектов с машины на машину. На первый взгляд, это положение кажется странным, так как объект в парадигме ООП инкапсулирует внутри все свои свойства и выглядит как агрегат, который может легко монтироваться и демонтироваться из пространства объектов. Фактически же объект имеет цепочки связей с вычислительной системой, что потребует его «упаковку» в пакет на месте создания и «распаковку» в другой системе на месте использования. Для таких действий необходима стандартизация протокола пакета и самого языка. В настоящее время эти важные задачи не решены. В случае решения этих проблем программирование приблизится к промышленному проектированию, когда конструктор создает проект машины, используя готовые агрегаты и узлы и синтезируя недостающие части.

Способы синтезирующего программирования в объектно-ориентированной системе были освещены в предыдущих разделах достаточно подробно при рассмотрении иерархии классов и механизма наследования, а также инструментально-интерфейсных средств. Повторно отметим, что SmallTalk проектировался таким образом, чтобы среда программирования способствовала успешному творчеству человека за пультом компьютера.

Выводы

Притча о пользе проектирования сверху вниз. Симптоматической является ситуация, когда программирование как наука развивается снизу вверх. Для демонстрации этого можно использовать схему уровней абстракций, предложенную Дейкстрой [30]. I-й уровень абстракции представлен абстрактной машиной M_i. Программа уровня i может транслироваться в программу уровня i —1. Выполнение первой программы на машине M_i адекватно выполнению второй программы на машине M_{i-1}. На самом нижнем уровне реальная машина M_0 интерпретирует программу в машинных кодах. На самом верхнем уровне абстракции на недостигаемой высоте, по-видимому, находится человеческий интеллект, так как человек всегда будет стремиться сделать ЭВМ равноценным партнером.

Если проектирование машины следующего уровня абстракции ведется с ориентации на возможности машины более низкого уровня без глубокого учета человеческого фактора, т. е. снизу вверх, то это приводит к ошибкам и заблуждениям, которые, к сожалению, тиражируются миллионными экземплярами не только в виде книг и программ. Яркий пример того — история с АСУП.

SmallTalk является положительным примером, когда при разработке ставилась задача обеспечить поддержку воображаемого мира пользователя с целью повышения его творческих способностей при работе за компьютером [31]. Это движение происходило интуитивно, почти вслепую, но в результате было открыто новое направление в программировании.

По-видимому, программирование достигло такого уровня, что для дальнейшего зримого движения вверх необходимо попытаться спроецировать законы психологии на ПО. Этот процесс имел бы направленность сверху вниз в рассмотренной схеме Дейкстры.

Наконец, хочется верить, что именно в таком направлении удастся решить целевую проблему, поставленную в работе [8, с. 101]:

«Все, что могут сделать профессиональные программисты для решения центральной задачи информационной технологии 80-х годов — формализации знаний — это попытаться создать типовую технологию (или спектр типовых технологических приемов, например по основным проблемным областям) для автоформализации знаний, т. е. разработать инструментальные средства, облегчающие непрограммирующим профессионалам процесс самостоятельной формализации их индивидуальных знаний».

ПРИЛОЖЕНИЕ.

Словарь ключевых терминов ООП

Объект	— компонент системы, представленный приватной памятью и набором операций.
Сообщение	— запрос объекту на выполнение одной из его операций.
Метод	— описание того, как выполнять одну из операций объекта.
Класс	— описание группы подобных объектов.
Экземпляр	— один из объектов, описываемых классом.

ЛИТЕРАТУРА

1. Byte.— 1981.— August, N 8.
2. Byte.— 1986.— August, N 8.
3. OOPSLA'86, conf. proc. // Sigplan notices.— 1986.— N 11.
4. OOPSLA'87, conf. proc. // Sigplan notices.— 1987.— N 12.
5. Verity J. W. The OOPS Revolution. // Datamation.— 1987.— May, N 1.— P. 73—78.
6. Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации. // Микропроцессорные средства и системы.— 1986.— № 1.
7. Громов Г. Р. Автоформализация профессиональных знаний. // Микропроцессорные средства и системы.— 1986.— № 3.
8. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации.— М.: Наука, 1984.
9. Goldberg A. and Robson D. SmallTalk-80: The Language and Its Implementation.— Addison-Wesley, Reading, Mass. 1983.
10. Goldberg A. SmallTalk-80: The Interactive Programming Environment.— Addison-Wesley, Reading, Mass. 1983.
11. Krasner G. Smalltalk-80: Bits of history, Words of advice.— Addison — Wesley, Reading, Mass. 1983.
12. Schmucker K. J. Object — oriented languages for the Macintosh. // Byte.— 1986.— August.— P. 177—185.
13. Davis M. Smalltalk/V. Release 1.2. // Byte.— 1987.— June.— P. 256—262.
14. Ingalls D. Yesterday's horizon. // HOOPLA!.— 1988.— V1, N 2.
15. Akinori Yonezawa and Yoshihiro Matsumoto, Object Oriented Concurrent Programming and Industrial Software Production // LNCS 186: Formal Methods and Software Development. Proceeding.— 1985.— V2.— P. 395—409.
16. Майерс Г. Надежность программного обеспечения.— М.: Мир, 1980.
17. Фокс Д. Программное обеспечение и его разработка.— М.: Мир, 1985.
18. Брукс Ф. П. мл. Как проектируются и создаются программные комплексы. Мифический человеко-месяц.— М.: Наука, 1979.
19. Tesler L. The Smalltalk Environment // Byte.— 1981.— August, N 8.— P. 90—147.
20. Флейшман Б. С. Системология, системотехника и инженерная экология. Кибернетика и ноосфера.— М.: Наука, 1986.
21. Майерс Г. Архитектура современных ЭВМ.— М.: Мир, 1985.
22. Abbott R. J. Knowledge abstraction // Comm. ACM.— 1987.— V30, N 8.— P. 664—671.
23. Cox B. J. Message/Object Programming: An Evolutionary Change in Programming Technology // IEEE SoftWare.— 1984.— Junuary.— P. 50—61.
24. Pascoe G. A. Elements of object-oriented programming // Byte.— 1986.— August.— P. 139—144.
25. Требования и спецификации в разработке программ. Сборник статей.— М.: Мир, 1984.
26. Mallett M. Advantage C++ and Guidelines C++. // Byte.— 1987.— October.— P. 229—234.
27. Enderle G., Kansy K., Pfaff G. GKS — The Graphical Standard.— Berlin — Heidelberg — New York, Springer, 1983.
28. Goldberg A. Programmer as reader // Proc. conf. "Information processing 86", Elsevier Science Publishers B. V. (North — Holland), 1986.
29. Ершов А. П. Предварительные соображения о лексиконе программирования. Сб. Кибернетика и вычислительная техника, Выпуск 1, М.: Наука, 1985.
30. Dijkstra E. W. Notes on Structured Programming, Structured Programming, (Eds. O. J. Dahl, et al), Academic Press, 1972.
31. Goldberg A. Introducing the Smalltalk-80 system. // Byte.— 1981.— N 8, August.— P. 14—26.

152140, Переславль-Залесский
Ярославской обл., ИПС АН СССР,
Крюков В. А.

Статья поступила 25.10.88

УДК 681.3

С. М. Абрамов, С. П. Пименов, А. А. Абакумов, М. И. Хаткевич

КОМПЬЮТЕРНЫЙ ВИРУС

Одной из причин, вызывающих сбои в нормальном функционировании вычислительных систем, являются несанкционированно репродуцирующиеся программы, называемые компьютерными вирусами. Результат их деятельности на компьютере ведет к потере информации, нарушению работы прикладного и системного программного обеспечения.

Вирус проникает на компьютер вместе с программами-вирусоносителями. При этом зараженная программа внешне и функционально ничем не отличается от других программ. Определить и выявить вирус на ранней стадии его появления в системе можно только специальными средствами. Зачастую пользователь замечает его, когда компьютер уже серьезно поражен и определенная часть информации безвозвратно потеряна. Особенно опасен вирус для конечного прикладного пользователя, не обладающего квалификацией системного программиста, что характерно для многих пользователей персональных ЭВМ.

Правдивая история из жизни вируса DOS-62

Эта история произошла в одном из академических институтов в середине августа 1988 г.

Как было замечено еще в начале компьютерной эры, время от времени ЭВМ начинают вести себя странно, но тогда ЭВМ были крайне ненадежны. Странное поведение такой надежной техники, как IBM PC XT, вызывает удивление, а IBM PC XT вела себя странно.

Первой с этим столкнулась старший техник лаборатории. К ее жалобам на ЭВМ сначала не отнеслись серьезно. И с удовольствием взялись объяснять симпатичной девушке, как нужно обращаться с ЭВМ.

Но первые же действия показали, что не все благополучно. Попытки запустить программу приводили к тому, что ЭВМ выходила на перезагрузку. Это еще не вызвало опасений. Такие случаи бывают и на IBM PC XT, когда портится информация на магнитных носителях. Испорченная программа не была редкостью, и ее восстановили с архивных дискет. Но ситуация запутывалась еще больше. Произошло нечто удивительное. После перезагрузки ЭВМ задумывалась и через некоторое время снова выходила на перезагрузку.

Под подозрением оказались системные программы COMMAND.COM,AUTOEXEC.BAT, драйверы, указанные в CONFIG.SYS. Перезагрузка с дискет проходила нормально. Это позволило провести ряд экспериментов. Вставленные в AUTOEXEC.BAT команды выдачи сообщений на экране не появились. Постепенно снимались подозрения и с драйверов.

Наконец причина была найдена. Дело было в программе COMMAND.COM. Это было заметно даже невооруженным взглядом. Размеры COMMAND.COM отличались от размеров эталонного варианта программы: были больше примерно на 0,5 Кбайт. Исправить ошибку не представляло никакого труда, COMMAND.COM был скопирован, и, казалось, инцидент был исчерпан.

Но тут произошло событие, которое заставило очевидцев усомниться в реальности происходящего. Размеры по-

вого COMMAND.COM увеличились буквально на глазах.

«Этого не может быть!»

Но COMMAND.COM стал таким же, каким был раньше. Такое поведение ЭВМ было воспринято, как вызов.

«Да как она смеет не слушаться!»

И испорченный COMMAND.COM был запущен под отладчиком. Первые же команды показали наличие злого умысла, а не случайной ошибки.

После бурного обсуждения была выработана программа действий. Стало ясно, что это вирус. Тот самый, о котором несколько раз писали в газетах, но на последних страницах в разделе курьезов. И даже многомиллионные суммы ущерба, нанесенные вирусом вычислительным центрам где-то там, в капиталистических странах, не воспринимались серьезно.

Теперь нам самим пришлось столкнуться с тем грустным фактом, что на наших ЭВМ, где хранятся сотни файлов — результаты работы многих людей в течение долгого времени, хозяйничает какая-то программа, которая неизвестно, как сюда попала, и неизвестно, что здесь делает, что уже испортила и как с ней бороться. Мы почувствовали себя в роли микробиологов, столкнувшихся с эпидемией неизвестного вируса. И хотя нашими пациентами были компьютерные программы, а не живые существа, такие аналоги оказались весьма близкими.

Эксперименты заключались в поиске следов вируса — составляющих его «аминокислот». Первые же попытки найти на диске константы из тела вируса, показали, что около сотни программ их содержат. Более детальное рассмотрение установило, что это не случайные совпадения, после чего был объявлен строжайший карантин. Все дискеты, с которых работали на ЭВМ, защищены, программы не переносились между ЭВМ.

Затем проделали работу по «дезинфекции» диска — тщательно уничтожили все программы, содержащие следы вируса*.

Эксперименты над вирусом были продолжены «в пробирке» — отдельной директории с «отсаженным» «вирусоносителем». Вирусу «подсаживались» подопытные кролики — программы, и исследовались способы распространения инфекции.

После чего проведена работа по выведению «чистой культуры» вируса. Получена программа, содержащая только текст вируса. Детальное исследование чистой культуры вируса под микроскопом — отладчиком, позволило составить «генетическую карту» вируса — тщательно прокомментированный реассемблированный текст программы-вируса.

Дальнейшие работы были направлены на создание средств противодействия эпидемии, включающие диагностику факта заражения, обеззараживание — уничтожения вируса, и восстановление пораженных вирусом файлов. Методы создания таких средств найдены в процессе исследования вируса. Создана программа ANTIVIRUS, которая реализовала такие средства. В программе ANTIVIRUS использованы алгоритмы вируса, т. е. он сам послужил основой для создания вакцины.

Затем проведена кампания по дезинфекции всего программного обеспечения в институте как на винчестерах персональных ЭВМ, так и на гибких дисках. Путь проникновения вируса в институт так и остался не выяснен. Естественно, что он был занесен когда-то с одной из программ. Однако, когда было выявлено наличие вируса, существовали сотни зараженных программ. И невозможно было определить, откуда и как распространялось заражение.

Нам не известны какие-либо сообщения именно об этом вирусе. Поэтому мы посчитали себя вправе дать ему имя DOS-62.

По нашим оценкам, данная версия вируса была достаточно гуманной. Возможны значительно более коварные мутации данного вируса, которые могут быть получены добавлением всего нескольких команд. Будем надеяться, однако, что этого никогда не произойдет.

Характеристика вируса DOS-62

Вирус DOS-62 — маленькая программа, написанная высококвалифицированным программистом, которая обладает повышенной живучестью и мобильностью. Вирус запускается программой-носителем и во время своей работы действует следующим образом. Он ищет жертву, и если таковая находится, то с вероятностью семь восьмых она будет инфицирована; с вероятностью одна восьмая файл-жертва будет модифицирован так, что дальнейшее его использование и восстановление станет невозможным. Завершив свою работу, программа-вирус передает управление программе-носителю, которая успешно осуществляет предписанные ей действия. Таким образом, инфицированные программы выполняются нормально и пользователь не сразу обнаруживает опасность. Наш горький опыт показывает, что проходит от одного до трех месяцев (в зависимости от интенсивности работы), прежде чем появляются серьезные основания думать о том, что с машиной творится что-то неладное.

Такое поведение вируса говорит прежде всего о его стремлении заразить как можно больше программ и тем самым распространиться как можно шире, обеспечить свою выживаемость. DOS-62 с одинаковой легкостью путешествует с винчестера на гибкие диски и обратно, а на гибких дисках — с машины на машину, из города в город, из страны в страну. Возможно распространение DOS-62 по сетям.

Причины, способствующие распространению вирусов

С каждым годом парк вычислительных машин увеличивается и расширяется. Особенно бурный рост наблюдается в области персональных компьютеров. Все большее число пользователей-непрофессионалов вовлекается в работу с компьютером. В этих условиях заражение машины вирусом очень опасно, так как рядовой пользователь, как правило, не в состоянии самостоятельно побороть эту компьютерную болезнь.

В нашей стране положение усугубляется тем, что пока плохо развиты правовой институт по защите авторских прав на ПО и товарно-денежные отношения на рынке ПО.

Существующая практика натурального обмена программами (особенно широко распространенная в мире персональных компьютеров), при которой никто ни за что не несет ответственности, является благодатной питательной средой для разного вида вирусов.

Развитие сетевых средств общения компьютеров, как показывает зарубежный опыт, несет с собой еще большую угрозу компьютерному миру.

Другая важная сторона обсуждаемого явления — вопрос о мотивах разработчика и распространителя вируса. По нашему мнению, он либо не осознает последствия своей деятельности, либо страшно зол на весь мир и хочет ему досадить. В любом случае, ему следует точно представлять, что большинство его жертв — ни в чем неповинные люди. А поэтому данное явление может быть квалифицировано как слепой компьютерный терроризм, не признающий границ, городов и государств, карающий (за что?) всех подряд и больнее всего тех, кто слабее.

Есть ли спасение?

У профессионала-программиста столкновение с вирусом вызывает бурю эмоций в первый момент. И исход схватки, как правило, предрешен. Но последствия не всегда однозначны. В любом случае, это потерянное на борьбу с вирусом время, потерянные файлы, «откат» в выполняемых плановых разработках.

Что делать простым пользователям? Для них необходимы эффективные средства защиты, диагностики, лечения.

По степени сложности антивирусные средства условно можно расположить в следующем порядке:

* Как показали дальнейшие работы, этого можно было и не делать, если бы мы имели в то время замечательную программу — ANTIVIRUS *(Прим. автора)*.

диагностика определенного вируса;
лечение либо уничтожение пораженных файлов;
предохранение (защита) от вирусной атаки.

В настоящее время известны пока простые средства, направленные на борьбу с конкретными разновидностями вируса. Разработка надежных и универсальных средств защиты требует времени и средств. Профессиональная сервисная служба по защите ПЭВМ от вирусов только зарождается.

Что касается вируса DOS-62, то его судьба предрешена. В настоящее время при Ленинградском кооперативе «Терм» организуется подразделение по сервисному обслуживанию ПЭВМ. Одной из услуг, оказываемой данной бригадой, будет диагностика и устранение программ-вирусов с машинных носителей IBM PC XT/AT. Для этого будет использована последняя версия программы ANTIVIRUS, позволяющая провести абсолютно надежное диагностирование и лечение IBM PC XT/AT от трех вирусов, распространенных в СССР: DOS-62, «листопад» и «черная дыра». Бригада выполняет сервисное обслуживание организаций и «индивидуалов» по диагностике и лечению ПЭВМ IBM PC XT/AT от этих вирусов. Если Вы будете регулярно (по мере приобретения извне подозрительных файлов) получать сервисное обслуживание, то указанные вирусы для Вашей ПЭВМ не страшны.

Разбор принципов функционирования вирусов DOS-62, «листопад», «черная дыра» и других позволил нам построить практически полную классификацию возможных вирусов и средств борьбы с ними. Реализован комплекс программ PC Shield, защищающий ПЭВМ почти от всех теоретически возможных вирусов. Решается вопрос о поставке пакета PC Shield пользователям.

И мы верим, что пользователь будет защищен.

Телефон 359-37-80, Москва

Статья поступила 20.10.88

БАЗЫ ДАННЫХ

УДК 681.3

В. К. Акулов, В. И. Журавлев, Н. А. Новожилов, С. Л. Шинкевич

СИСТЕМА УПРАВЛЕНИЯ РЕЛЯЦИОННЫМИ БАЗАМИ ДАННЫХ

Существующий дефицит отечественного прикладного программного обеспечения, ориентированного на массового пользователя микроЭВМ и персональных компьютеров, в некоторой степени сдерживает широкое внедрение последних в различных областях народного хозяйства и научных исследованиях. Наиболее остро эта проблема стоит в области хранения и управления данными на машинах такого класса.

К настоящему времени разработано и находится в эксплуатации большое количество систем управления базами данных (СУБД), обладающих развитыми языками описания и управления данными, например, таких как БАНК, ОКА, СЕТОР, СЕДАН и др. В основном эти системы ориентированы на большие ЭВМ и требуют значительных машинных ресурсов, в частности, наличия оперативного и внешнего запоминающихся устройств большой емкости. Кроме того, эти СУБД, как правило, сложны в эксплуатации для неподготовленного пользователя.

В большинстве случаев конечный пользователь ПЭВМ не имеет специальных знаний о принципах построения и ведения баз данных (БД). Поэтому, очевидно, программное обеспечение поддержки БД должно предоставлять пользователю простой, понятный и наглядный интерфейс взаимодействия с данными, включающий функционально полный и гибкий язык манипулирования данными.

Из традиционных подходов к организации данных наиболее полно этим требованиям отвечает реляционный. Он основан на том, что наборы данных, удовлетворяющих определенным ограничениям, рассматриваются как математические отношения, и для решения некоторых практических задач, связанных с обработкой таких наборов данных, может использоваться элементарная теория отношений. В реляционной БД отношение можно представить как обычную двумерную таблицу, с которой удобно и привычно работать. Немаловажно и то, что пользователю не нужно помнить пути доступа к данным (как в иерархической и сетевой моделях) и строить соответствующие алгоритмы обработки запросов. На языке реляционного исчисления в запросе указываются только требования к формированию нового отношения, а не способ его получения.

Предлагаемая СУБД RDBMS (The Relational Data Base Manasement System), разработанная в СКБ НП ОПТИКА Томского филиала Сибирского отделения АН СССР, представляет собой реализацию реляционного подхода в хранении и управлении данными на ЭВМ типа «Электроника 60», «Электроника МС 1212.02», СМ4, «Электроника 85» и других машинах этого класса. Система RDBMS позволяет создавать и вести несколько различных БД. Она предоставляет возможность коллективного доступа к данным многих пользователей с различных терминалов, а также из прикладных программ, составленных на языках программирования высокого уровня и Макроассемблере. RDBMS работает под управлением ОС RT-11 или TSX—Plus и занимает не более 40 Кбайт оперативной памяти.

Структура базы данных

С точки зрения структуры реляционная БД — это набор о т н о ш е н и й, представляемых пользователем в виде двумерных таблиц, в которых столбцы называются а т р и б у т а м и, а строки — к о р т е ж а м и. RDBMS накладывает ограничения на размер отношений: число атрибутов в любом отношении не может быть больше 32 (при соответствующей генерации — 64), а кортежей — не более 65535. Как правило, эти ограничения не вызывают каких-либо трудностей в конкретных приложениях.

Для каждого атрибута задается множество значений, которые могут попадать в этот атрибут. Такое множество называется д о м е н о м. RDBMS предоставляет пользователю набор из восьми основных доменов. Путем их ограничения можно определять свои домены и присваивать им имена. Ниже приведены имена основных доменов и характеристики типов данных, которые относятся к ним:

INTEGER — целое число в диапазоне —32767...+3267;
UNSIGN — целое положительное число в диапазоне 0...+65535;
LONG — целое число в диапазоне —2147418112...+2147483647;
REAL — положительное или отрицательное вещественное число с абсолютным значением в диапазоне 0.29E—38...1.7E38;
CHAR — один символ в коде КОИ-8, один байт;
STRING — текстовая строка, содержащая не более 80 символов;
FILE — текстовый или любой другой двоичный файл, байтовый массив неограниченной длины;
DATE — календарная дата.

Отношения БД связаны между собой значениями данных общих атрибутов одного и того же домена. Такие атрибуты называются внешними ключами. Существуют также внутренние ключи — это те атрибуты, которые однозначно идентифицируют кортеж внутри отношения. Никаких два кортежа не могут иметь одинаковых значений в ключевых атрибутах. Как внешних, так и внутренних ключей в отношении может быть несколько. Отношения подчиняются четырем правилам: не может быть двух совпадающих кортежей; порядок кортежей безразличен; несколько атрибутов в одном или различных отношениях могут быть определены на одном и том же домене, но внутри одного отношения имя каждого атрибута должно быть уникально; порядок следования атрибутов в отношении может быть любой.

Организация физического хранения данных

Отношения реляционной БД располагаются на внешних носителях информации, которыми могут быть любые поддерживаемые ОС устройства с произвольным доступом и каталоговой организацией. Каждое отношение хранится в виде трех физических файлов, размеры которых динамически изменяются в зависимости от наполнения отношения данными. Файлы отношения имеют одинаковые имена, совпадающие с именем отношения, но различаются расширением.

Кортежи, хранимые в одном из файлов (файл кортежей),— это цепочка данных. Порядок следования данных жестко связан с порядком атрибутов в отношении. В поле кортежа, соответствующем домену STRING или FILE, всегда помещается указатель на данные, которые хранятся в другом файле (файле данных). Третий (индексный) файл содержит указатели на кортежи в кортежном файле. Порядок индексов задает порядок кортежей в отношении. При операции исключения кортежа из отношения кортеж не удаляется, а удаляется только его индексная метка.

В файле кортежей имеется каталог пользователей, который содержит список паролей и имен пользователей, имеющих право доступа к отношению, и определены привилегии каждого из них.

Обычно поиск каких-либо данных в отношении осуществляется путем последовательного просмотра всех кортежей. При быстром поиске данных используется механизм кеш-адресации. Для одного или нескольких атрибутов предварительно создаются специальные кеш-таблицы в индексном файле, с помощью которых в дальнейшем по значениям самих данных определяется позиция искомого кортежа в отношении.

Интерактивный интерфейс пользователя

При разработке СУБД RDBMS большое внимание было уделено интерфейсу пользователя. Он представляет собой непроцедурный язык формулирования запросов к БД. Язык реализован в виде набора команд и схож с командным языком DCL ОС RT-11. В нем допускаются сокращения ключевых командных слов и аргументов. Для начинающих осваивать СУБД RDBMS наиболее удобна форма ввода команд с использованием подсказок. При этом требуется набрать на терминале лишь идентификатор команды. Система будет задавать наводящие вопросы,чтобы пользователь указал недостающие аргументы для полного формулирования запроса. Кроме того, к услугам пользователя имеется справочник языка, встроенный в систему, который может выдать список допустимых команд RDBMS, а также информацию о синтаксисе, допустимых ключах любой команды и примеры их применения.

Все команды языка можно условно разбить на четыре группы. К первой группе относятся команды, оперирующие с отношениями. Команда CREATE используется для создания нового отношения базы данных. Она предлагает пользователю заполнить графы в специальном экранном формуляре, который описывает отношение. Пользователь присваивает имя отношению и всем составляющим его атрибутам, а также специфицирует домены каждого атрибута.

Такой же формуляр вызывается командой MODIFY, но он позволяет модифицировать уже существующее отношение. Здесь можно удалить некоторый атрибут из отношения или добавить новый, переименовать атрибут, изменить домен.

Команда DELETE удаляет отношение из БД.

Формуляр MANAGER дает возможность администратору БД определить доступ к отношениям 15 пользователям. Каждому пользователю задаются индивидуальные привилегии доступа к данным. Администратор может запретить операции удаления или добавления кортежей, указать атрибуты, запрещенные для модификации в них данных. Пользователям можно определить индивидуальные модели отношения путем перечисления доступных атрибутов, к другим атрибутам отношения любой доступ будет запрещен.

Перечисленные выше команды являются исключительной привилегией администратора БД. Кроме них к этой группе относятся и другие команды. Команда SORT сортирует строки таблицы по данным любого атрибута в восходящей или нисходящей последовательности.

Две таблицы, имеющие общий внешний ключ, можно соединить в одну командой JOIN. Командой PROJECT можно любые выбранные атрибуты переслать в другое отношение, а командой CORY в другое отношение пересылаются только выбранные кортежи.

Ко второй группе относятся команды, с помощью которых можно модифицировать данные внутри отношения. Команды INSERT и REMOVE позволяют соответственно вводить новые строки в таблицу или удалять их. Команда UPDATE производит локальную замену данных в указанном поле одного кортежа, а команда REPLACE — поиск и замену данных во всех выбранных кортежах.

Связь между БД и файлами ОС обеспечивает команда SEND. Она пересылает данные из отношения в файл и обратно.

Команда CALCULATE выполняет арифметические операции с данными в указанных столбцах таблицы.

К третьей группе следует отнести команды выборки и просмотра данных. Команда SELECT позволяет задавать сложные логические условия выборки данных в отношении. Указатель текущего кортежа изменяют командами NEXT и SEARCH.

Просмотр данных текущего кортежа осуществляется с помощью команды LOOK, а всех выбранных строк отношения — командой LIST. LIST создает текстовую таблицу заданного формата, которую можно вывести на экран терминала, печатающее устройство или в текстовый файл.

В четвертую группу входят сервисные команды:

DIRECTORY	— распечатка каталога отношений на устройстве;
HELP	— выдача справочной информации о командах языка;
LOGON	— регистрация пользователя;
SET	— установка контрольных системных параметров;
SHOW	— просмотр контрольных параметров СУБД и другой вспомогательной информации.

В СУБД RDBMS имеются средства определения макроязыка на базе основных команд. Это позволяет определить свой синтаксис команд, построить меню, т. е. создать проблемно-ориентированный язык взаимодействия с БД. RDBMS располагает также средствами загрузки данных в отношения из специально подготовленных файлов.

Программный интерфейс к базе данных

Для создания специализированных систем обработки информации в RDBMS включен программный интерфейс. Он предоставляет возможность включить функции доступа

к БД в прикладные программы, написанные на любом языке программирования. Программный интерфейс реализован в виде набора программных запросов из специальной библиотеки на этапе сборки пользовательской программы. В библиотеку включены функции открытия и закрытия отношений для доступа, добавления и удаления кортежей, элементарные функции чтения и записи данных, а также ряд сервисных функций.

Базовая версия программного интерфейса ориентирована на такие языки программирования, как Паскаль, Си, Фортран и PL-11. Однако пользователь-программист может применить специальный модуль для описания способа передачи аргументов при вызове функций из любого другого языка программирования.

Многотерминальный режим работы

Режим коллективного доступа к БД можно организовать на достаточно мощных ЭВМ типа СМ4, «Электроника МС 1212.02», которые оснащены ОС TSX-Plus. RDBMS не имеет реентерабельного ядра, поэтому каждый пользователь работает со своей копией СУБД в памяти.

Вопрос разрешения конфликтных ситуаций при модификации отношений БД решается путем временного захвата отношения в монопольное использование. Захватить отношение может только тот пользователь, который начал первым производить обновление данных. Захваченное отношение помечается специальным кодом пользователя в одном из трех файлов отношения. Алгоритм захвата включает в себя следующие этапы. Сначала проверяется, свободно ли отношение, т. е. отсутствие кода другого пользователя. Если оно свободно, то производится запись кода, идентифицирующего пользователя. После этого, если при чтении статуса отношения возвращается этот же код, то захват отношения считается успешным. Однако такой подход к решению проблемы может доставить некоторые неприятности при сбоях в работе системы, так как при перезагрузке могут остаться захваченные отношения.

Наличие у пользователей широкого парка терминалов, отличающихся друг от друга некоторыми функциональными характеристиками, может вызывать несовместимость с базовым вариантом СУБД, которая поддерживает терминал типа VT-52. В такой ситуации пользователь может на этапе генерации СУБД в специальном файле описать основные функции своего терминала и, если необходимо, назначить различные типы терминалов на разные физические линии своего компьютера.

Области применения RDBMS

RDBMS наиболее эффективна в тех случаях, когда информацию можно представить в табличной форме. В настоящее время имеется несколько вариантов применения системы RDBMS в различных приложениях, связанных с хранением и обработкой данных. Например, уже в течение двух лет эксплуатируется на базе СУБД RDBMS система «Склад» по учету и комплектации электрорадиоэлементов. Система включает в себя БД, состоящую из нескольких отношений, и управляющую программу, оформленную в виде меню.

Система «Склад» построена так, что пользователю в ответ на затребованные комплектующие выдается информация о их наличии, и если таких комплектующих нет, то предлагается возможная их замена. В этом списке пользователь указывает требуемое число компонентов и получает информацию об их местонахождении (номер тары). Информация о всех актах изъятия комплектующих из склада фиксируется в специальном отношении, по данным которого в дальнейшем производится контроль расхода радиоэлементов. Кроме этих функций система позволяет получать различного рода справочную информацию о любых радиоэлементах, например тип корпуса, технические условия, содержание драгоценных металлов, цену и др.

Другим примером применения RDBMS может служить разработанная совместно с НИИ онкологии Томского научного центра АМН СССР система анализа онкологических данных (The Breast Cancer Data Analysis System). Основой ее служит БД, заполненная различной информацией о пациентах. Сюда входят как анамнестические данные, так и данные многочисленных лабораторных исследований. Система оснащена мощным программным аппаратом статистической обработки, позволяющим производить анализ данных у больных по различным группам и стадиям заболевания. Подсистема корреляционного анализа позволяет выявить взаимосвязь между различными параметрами сложных динамических систем внутренней среды организма человека. В результате можно не только получать оперативную информацию о состоянии пациента, но и прогнозировать течение заболевания и его исход.

Результаты статистической обработки выводятся в виде графиков, гистограмм на экран графического дисплея и на графическую печать. Система ориентирована на пользователя-врача и поддерживает диалог в форме меню.

Созданы и с успехом применяются также и другие БД различного назначения. В настоящее время СУБД RDBMS эксплуатируется примерно в 20 организациях страны. Опыт ее использования позволяет утверждать, что разработанная СУБД обладает достаточной универсальностью, простотой, компактностью.

Телефон 1-86-33, Томск

Статья поступила 14.07.87

УДК 681.03.06

Д. К. Волков

МИКРОСУБД НА ЯЗЫКЕ БЕЙСИК ДЛЯ ПЭВМ

Основные свойства СУБД для ПЭВМ — обеспечение программно-информационной и машинной независимости.

При настройке СУБД на конкретные приложения должны быть заданы тип каждого данного (целое, вещественное, двоичное, символьные), длина, а также состав и расположение данных в логической записи.

Для достижения истинной независимости [1] необходимо, чтобы СУБД не требовалось специально настраивать на конкретные свойства данных.

Этого можно добиться, если представлять все данные в символьном виде и группировать их в логические записи на этапе считывания-записи, идентифицируя принадлежность к логической записи по уникальному коду записи и порядковому номеру физической записи в сегменте.

Использование языков высокого уровня [2] обеспечивает машинную независимость СУБД. Для ПЭВМ целесообразно применять наиболее распространенный язык — БЕЙСИК. В языке БЕЙСИК символьные данные имеют неопределенную длину, поэтому можно организовать считывание записей следующим образом:

(1)

```
FOR K=1 TO R1
INPUT #1,D1$(K)
NEXT K
```

где R1 — число данных логической записи, задается в момент запуска программы управления базой данных.

Преобразование символьного представления числового данного в вещественный вид осуществляется командой VAL:

```
10 F1=SYS(7,0)
20 REM        НАЗВАНИЯ РЕКВИЗИТОВ
30 Y=TTYSET(255%,80)
40 R3%=50
50 DIM M1$(100)
52 DIM D1$(70%) \ REM      ПОЛЕ ДЛЯ РАЗМЕЩЕНИЯ ЗАПИСИ БАЗЫ
60 M1$(1)='1,ФАМИЛИЯ И.О. '
70 M1$(2)='2,ПОЛ '
80 M1$(3)='3,ДАТА РОЖДЕНИЯ '
90 M1$(4)='4,МЕСТО РОЖД '
100 M1$(5)='5,НАЦИОНАЛЬНОСТЬ '
110 M1$(6)='6,ПАРТИЙНОСТЬ '
120 M1$(7)='7,ОБРАЗОВАНИЕ '
130 M1$(8)='8,УЧ.ЗАВЕДЕНИЕ'
140 M1$(9)='9,СПЕЦИАЛЬНОСТЬ '
150 M1$(10)='10,УЧ.СТЕП. УЧ.ЗВАН '
160 M1$(11)='11,ГОД ПОС.НА ПРЕДП '
170 M1$(12)='12,ОСН.ПРОФЕССИЯ '
180 M1$(13)='13,РАБ.В ОТРАСЛИ '
190 M1$(14)='14,ДОЛЖНОСТЬ '
200 M1$(15)='15,СЕМ.ПОЛОЖ '
210 M1$(16)='16,КОЛ.ДЕТЕЙ '
220 M1$(17)='17,ПООЩРЕНИЯ '
230 M1$(18)='18,ТАБЕЛЬНЫЙ НОМЕР '
280 M1$(R3%+2)='М  Ж' \ M1$(R3%+3)='ГГММЧЧ' \ M1$(R3%+11)='67 75 ....'
290 M1$(R3%+5)='РУС ТАТ УКР БЕЛ ...' \ M1$(R3%+15)='С  Х'
300 M1$(R3%+7)='ВЫСШ ОЧН СРТЕХ ЗАОЧ' \ M1$(R3%+10)='КТН СНС ДТН ПРОФ...'
310 M1$(R3%+8)='ИФФИ 68  МГУ 75  ЛГУ  80...'
320 M1$(R3%+9)='ПРОГР СИСТЕМОТЕХН ЭЛЕКТРОН МАТЕМАТ ...'
330 M1$(R3%+14)='ЛАБ СТ.ЛАБ ИНЖ СТ.ИНЖ ВЕД.ИНЖ С.Н.С НАЧ.СЕК ...'
340 M1$(R3%+17)='БЛАГ ГРАМ ОРД МЕД' \ M1$(R3%+4)='МОСКВА КЛИН ВЛАДИМИР.ОБЛ...'
350 M1$(R3%+6)='ПР БП ВЛКСМ' \ M1$(R3%+20)='227-25-91'
360 REM ******          ЭКРАН-НАЗВАНИЕ                          ****** 1.1.
365 Z1=40 \ Z2=32 \ GOSUB 2000
380 PRINT TAB(10);'ИНТЕРАКТИВНАЯ ПРОГРАММА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ'
390 PRINT TAB(22);'(ИНФОРМАЦИЯ ПО КАДРАМ)'
400 N=9 \ GOSUB 2100
410 PRINT 'ИНСТИТУТ "ЦЕНТР",МОСКВА,1987'
411 PRINT '254-05-94 ВОЛКОВ Д.К.'; \ INPUT R1% \ R2%=R1%
430 OPEN 'LP:' FOR OUTPUT AS FILE #2
440 OPEN 'MX1:BAZA' FOR INPUT AS FILE #1
450 REM ****** ВВОД ПОИСКОВОГО  ПРЕДПИСАНИЯ С ЭКРАНА          ****** 1.2.
460 DIM D3$(30,3)
461 GOSUB 2000
470 PRINT TAB(10);'УКАЖИТЕ НОМЕР РЕЖИМА РАБОТЫ ПРОГРАММЫ:'
471 PRINT TAB(15);'ВЫДАЧА СОДЕРЖАНИЯ ИСКОМЫХ ДОКУМЕНТОВ          - 1'
472 PRINT TAB(15);'ПОДСЧЕТ КОЛИЧЕСТВА ИСКОМЫХ ДОКУМЕНТОВ         - 2'
473 PRINT TAB(15);'ВНЕСЕНИЕ ИЗМЕНЕНИЙ И ЗАПИСЬ НОВЫХ ДОКУМЕНТОВ - 3'
474 PRINT TAB(15);'ЗАВЕРШЕНИЕ РАБОТЫ С СИСТЕМОЙ                 - 4'
475 N=8 \ GOSUB 2100
500 INPUT P% \ C1=0 \ C2=0
501 IF P%>5 GO TO 470
510 ON P% GO TO 630,630,530,2420
520 REM ******     ПОДГОТОВКА РЕЖ 3                    (2)******2.1.
530 PRINT 'КОЛИЧ ДАННЫХ В РЕЗ МАССИВЕ'
531 T1=0
540 INPUT I2%
550 OPEN 'MX1:BAZA' FOR OUTPUT AS FILE #3
625 REM ****** ЗАДАНИЕ ПОИСКОВОГО ПРЕДПИСАНИЯ                ****** 1.3.
630 FOR K=1 TO R1%
631 GOSUB 2000
632 PRINT TAB(12);'ДОКУМЕНТЫ СОДЕРЖАТ СЛЕДУЮЩИЕ СВЕДЕНИЯ(ДАННЫЕ):'
640 PRINT  \ GOSUB 1100 \ PRINT  \ PRINT TAB(25);'СЛУЖЕБНЫЕ НОМЕРА:'
650 PRINT TAB(10)'99  - ЗАПУСК ПРОГРАММЫ НА ПОИСК ДОКУМЕНТОВ'
651 PRINT TAB(10)'100 - ОТКАЗ ОТ НАБРАННОГО ЗАДАНИЯ НА ПОИСК'
652 PRINT  \ PRINT 'УКАЖИТЕ  Н О М Е Р  ДАННОГО (ИЛИ  С Л У Ж Е Б Н Ы Й  НОМЕР),'
653 PRINT 'ПО ЗНАЧЕНИЮ КОТОРОГО ВЫ ХОТИТЕ ИСКАТЬ ДОКУМЕНТЫ'
654 PRINT
660 INPUT D3$(K,1)
670 IF D3$(K,1)='99' THEN K=R1% \ GO TO 740

671 IF D3$(K,1)='100' THEN K=0 \ GO TO 740
680 A1=VAL(D3$(K,1))
690 IF A1>R1% THEN 652
691 GOSUB 2000
692 PRINT TAB(25);M1$(A1) \ PRINT
700 PRINT TAB(10);'ДОКУМЕНТ МОЖНО ИСКАТЬ ОДНИМ ИЗ ЧЕТЫРЕХ СПОСОБОВ:'
701 PRINT '1 - СОДЕРЖАЩИЙ УКАЗАННОЕ  З Н А Ч Е Н И Е  ДАННОГО'
702 PRINT '2 - Н Е  СОДЕРЖАЩИЙ УКАЗАННОЕ ЗНАЧЕНИЕ ДАННОГО'
704 PRINT '3 - СОДЕРЖАЩИЙ ЗНАЧЕНИЕ ДАННОГО  Б О Л Ь Ш Е  ЗАДАННОЙ ВЕЛИЧИНЫ'
705 PRINT '4 - СОДЕРЖАЩИЙ ЗНАЧЕНИЕ ДАННОГО  М Е Н Ь Ш Е  ЗАДАННОЙ ВЕЛИЧИНЫ'
706 N=4 \ GOSUB 2100 \  \ PRINT 'УКАЖИТЕ  Н О М Е Р  НУЖНОГО СПОСОБА ПОИСКА '
707 INPUT D7%
708 IF D7%=1 THEN D3$(K,3)=' ' \ GO TO 720
709 IF D7%=2 THEN D3$(K,3)='НЕ' \ GO TO 720
710 IF D7%=3 THEN D3$(K,3)='Б' \ GO TO 720
711 IF D7%=4 THEN D3$(K,3)='М' \ GO TO 720
713 PRINT 'ОШИБКА !' \ GO TO 706
720 GOSUB 2000
721 PRINT TAB(25);M1$(A1)
722 PRINT TAB(15);M1$(R3%+A1)
728 N=10 \ GOSUB 2100
729 PRINT TAB(20);'УКАЖИТЕ КОНКРЕТНОЕ ЗНАЧЕНИЕ ДАННОГО
730 INPUT D3$(K,2)
740 NEXT K
750 P1=0
770 REM ******      СЧИТ.ОЧЕРЕД.ЗАПИСИ ДЛИНОЙ В R1% ДАННЫХ      ****** 1.4.
780 FOR K%=1% TO R1%
790 INPUT #1,D1$(K%)
810 NEXT K%
820 IF END #1 THEN RESTORE #1 \ P1=1
830 REM ******       ПРОВЕРКА ЗАПИСИ ПО ПРЕДПИСАНИЮ             ****** 1.5.
840 C1=C1+1
860 FOR K%=1% TO R2%
880 IF D3$(K%,1%)='99' THEN  GOSUB 1320 \ K%=R2% \ GO TO 990
890 D5%=0% \ R4%=VAL(D3$(K%,1%)) \ IF D3$(K%,3%)=' ' GO TO 950
895 IF D3$(K%,3%)='НЕ' THEN  GOSUB 1050 \ GO TO 960
900 IF D3$(K%,3%)='М' THEN  GOSUB 1260 \ GO TO 960
920 IF D3$(K%,3%)='Б' THEN  GOSUB 1290 \ GO TO 960
940 REM ******      ПРОВЕРКА ДЕСКРИПТОРА И ЗАПИСЬ ДЛЯ РЕЖ 3     (5)******2.7.
950 D5%=POS(D1$(R4%),D3$(K%,2%),1%)
960 IF D5%=0% THEN K%=R2%
970 IF P%<3% GO TO 990
980 IF D5%=0% THEN  GOSUB 1200
990 NEXT K%
1000 IF P%=3 THEN IF P1=1 GO TO 1580
1010 IF P1=0 GO TO 780
1020 GOSUB  2000
1022 PRINT TAB(10);'ПРОСМОТРЕНО ДОК-ТОВ:'C1'НАЙДЕНО ДОК-ТОВ:';C2;'ПРОЦЕНТ:'C2/C1*100
1023 N=11 \ GOSUB 2100
1024 PRINT 'НАЖМИ ЛЮБУЮ КЛАВИШУ'; \ INPUT W$
1030 GO TO 461
1040 REM ******      ОТРИЦАНИЕ ДЕСКРИПТОРА                       (5)******2.4.
1060 D5%=POS(D1$(R4%),D3$(K%,2%),1%)
1070 IF D5%=0% THEN D5%=1% \ GO TO 1090
1080 IF D5%=1% THEN D5%=0%
1090 RETURN
1100 REM ******    СОСТАВ РЕКВИЗИТОВ БАЗЫ                      (3)******2.2.
1110 FOR L=1 TO R1% STEP 4
1120 PRINT M1$(L);TAB(20);M1$(L+1);TAB(40);M1$(L+2);TAB(60);M1$(L+3)
1130 NEXT L
1140 RETURN
1200 REM ******      ЗАПИСЬ ДАННЫХ В BAZ                         (7)******3.1.
1210 FOR C%=1% TO 12%
1230 PRINT #3,D1$(C%)
1240 NEXT C%
1250 RETURN
1260 REM ******       МЕНЬШЕ ДАННОГО                             (5)******2.5.
1270 IF D3$(K%,2%)>D1$(R4%) THEN D5%=1%
1280 RETURN
```

```
R4=VAL(D1$(K))
```

(2)

Обратное преобразование для упаковки в базу важнейших результатов выполняется подпрограммой CHR¤:

```
D1$(K)=CHR$(R4)
```

(3)

Если в базе используются сегменты разной длины, число данных задается в начале сегмента нового типа и помечается знаком %. Конструкция (1) принимает следующий вид:

```
100 INPUT #,D1$(K)
102 A5$=SEG(D1$(K),1,1)
104 IF A5$='%' THEN A5$=SEG(D1$(K),2,4)\
    R1=VAL(A5$)\GOTO 108
106 GOTO 100
108 FOR K=1 TO R1
110 INPUT #1,D1$(K)
112 NEXT K
```

(4)

Пример

Входной набор данных:

```
1943
МОСКВА
К.Ф-М.Н.
%004
ЛУЗИН Б.Н.
622
250
НАЧ.СЕК.
%006
КОРОТКОВ П.П.
```

Результат работы фрагмента:

```
D1$(1)=ЛУЗИН Б.Н.
Д1$(2)=622
Д1$(3)=250
Д1$(4)=НАЧ.СЕК.
```

Истинная независимость данных для функции поиска:

```
200 FOR K=1 TO R1
202 IF D3$(K,1)="99" THEN GOSUB 300\
    K=R1\GOTO 210
204 D5+POS(D1$(D3(K,1)),D3$(K,2),1)
206 IF D5=0 THEN K=R1
208 NEXT K
210 RETURN
```

(5)

GOSUB 300 — подпрограмма отображения найденного документа.

Полный текст СУБД (Начало)

Полный текст СУБД (Окончание)

```
1290 REM ******        БОЛЬШЕ ДАННОГО                          (5)*****2.6.
1300 IF D3¤(K%,2%)<D1¤(R4%) THEN D5%=1%
1310 RETURN
1320 REM ******       РАСПЕЧ. НАЙДЕН. ДОК.  НА ЭКРАНЕ          (5)******2.3.
1330 C2=C2+1
1331 IF P%=3 THEN IF T1=4 GO TO 1550
1340 IF P%=2 GO TO 1560
1350 GOSUB 2000
1360 FOR K=1 TO R2%+1 STEP 2
1370 PRINT M1¤(K);TAB(20);D1¤(K);TAB(40);M1¤(K+1);TAB(60);D1¤(K+1)
1380 NEXT K
1390 N=3 \ GOSUB 2100
1400 IF P%<3 THEN  GOSUB 1740 \ GO TO 1560
1410 PRINT '1-УДАЛЯТЬ,2-КОРРЕКТИРОВАТЬ,3-ОСТАВИТЬ,4-ВЫЙТИ ИЗ РЕЖИМА'
1420 INPUT T1
1430 IF T1>4 THEN PRINT 'НОМЕР >4' \ GO TO 1400
1440 ON T1 GO TO 1560,1460,1550,1550
1460 REM ******                ЗАМЕНА ДАННЫХ       (2.3.)        ****** 3.2.
1470 PRINT 'НОМЕР ДАННОГО (5.5-ПЕРЕХОД К СЛЕД.ДОК-ТУ  6.6-ПОВТОР КОРРЕКТ.ДОК.'
1480 INPUT C
1490 IF C>I2% THEN PRINT 'НОМЕР БОЛЬШЕ';I2% \ GO TO 1460
1500 IF C=5.5 GO TO 1550
1510 IF C=6.6 GO TO 1350
1520 PRINT M1¤(C);TAB(20);M1¤(R3%+C)
1530 INPUT D1¤(C)
1540 GO TO 1470
1550 GOSUB 1200
1560 RETURN
1570 REM ******      ВВОД ДОКУМЕНТОВ В КОНЕЦ МАССИВА            ****** 1.6.
1580 PRINT 'НУЖЕН ВВОД ДОК-ТОВ(Д ИЛИ"ВК")'
1590 INPUT A¤
1600 IF A¤<>'Д' GO TO 1690
1610 FOR K=1 TO I2%
1620 PRINT M1¤(K);TAB(20);M1¤(R3%+K)
1630 INPUT D1¤
1640 PRINT #3,D1¤
1650 NEXT K
1660 PRINT 'ЕСЛИ КОНЕЦ ВВОДА,ОТВЕТЬТЕ:5.5; ИНАЧЕ '"ВК"'
1670 INPUT D4¤
1680 IF D4¤<>'5.5' GO TO 1610
1690 REM ******      ПЕРЕИМЕНОВАНИЕ ФАЙЛОВ                      ****** 1.7.
1700 CLOSE #3
1710 CLOSE #1
1730 GO TO 440
1735 REM ****** ПЕЧАТЬ ЭКРАНА РЕЖ 1,2: ПОИСК И СПРАВКА (2.3.)     ****** 3.3.
1740 PRINT 'ПЕЧ.ЭКРАНА-1,ОТК.ОТ ЗАПРОСА-2,ПРОДОЛЖЕН.ПОИСКА-3 РЕЖИМ СПРАВКИ-4'
1750 INPUT T1
1760 IF T1>4 GO TO 1740
1770 ON T1 GO TO 1790,1780,1870,1840
1780 P1=1 \ RESTORE #1 \ GO TO 1870
1790 FOR K=1 TO R2%+1 STEP 2
1800 PRINT #2,M1¤(K);TAB(20);D1¤(K);TAB(40);M1¤(K+1);TAB(60);D1¤(K+1)
1810 NEXT K
1820 PRINT #2 \ PRINT #2 \ PRINT #2
1830 GO TO 1740
1840 P%=2
1870 RETURN
2000 U1¤=CHR¤(72)
2010 U2¤=CHR¤(74)
2020 U3¤=CHR¤(27)
2030 U4¤=CHR¤(89)
2040 B¤=CHR¤(Z1)
2050 B1¤=CHR¤(Z2)
2060 PRINT U3¤+U1¤+U3¤+U2¤
2070 PRINT U3¤+U4¤+B¤+B1¤
2080 RETURN
2100 FOR K9=1 TO N \ PRINT  \ NEXT K9
2101 RETURN
2420 PRINT 'КОНЕЦ РАБОТЫ'
2440 END
2450 END
```

Значения поисковых дескрипторов:

K	D3(K,1)	D3(K,2)
1	2	622
2	4	НАЧ.СЕК.
3	99	

Содержание анализируемого документа:

```
D1$(I)

ЛУЗИН Б.Н.
622
250
НАЧ.СЕК.
```

Результат аналогичен фрагменту (4).

Номера (D3 ¤ (K, 1)) и значения поисковых дескрипторов вводятся пользователем в интерактивном режиме. С целью автоматизации процедуры задания дескрипторов поискового предписания программу можно дополнить таблицей названий дескрипторов:

```
DIM   M1$(100)
M1$(1)="1,ФАМИЛИЯ И.О."
M1$(2)="2,N СЕКТОРА"
M1$(3)="3.ОКЛАД"
M1$(4)="4,ДОЛЖНОСТЬ"
```

(6)

Истинная независимость для вывода результатов реализуется следующей подпрограммой:

```
FOR K=1 TO R1 STEP 2
PRINT #2,M1$(K);TAB(20):D1$(K);TAB(40);
M1$(K+1);TAB(60);D1$(K+1)
NEXT K
```

(7)

Подпрограмма предполагает ограничение длины D1 ¤ (K) и M1 ¤ (K) 20 символами и числа данных в сегменте 40...48 (при выводе на терминал).

Выходные наборы данных аналогичны фрагменту (6) и результату работы фрагмента (4).

Результат работы фрагмента:

```
1, ФАМИЛИЯ И.О.      ЛУЗИН Б.Н.
2, N СЕКТОРА        622
3. ОКЛАД              250
4, ДОЛЖНОСТЬ        НАЧ.СЕК.
```

Интерфейс с пользователем организован высвечиванием текста сверху вниз с помощью подпрограммы установки маркера с предварительной очисткой экрана:

```
U1$=CHR$(72)\U2$=CHR$(74)
U3$=CHR$(27)\U4$=CHR$(29)
B4=CHR$(32+Y)\G$=CHR$(32+X)
PRINT U3$+U1$+U3$+U2$
PRINT U3$+U4$+B$+G$;
RETURN
```

(8)

Здесь $y \leqslant 32$ — номер строки (сверху вниз), $x \leqslant 72$ — номер столбца (слева направо)

Доработки, проведенные в ходе опытной эксплуатации, и создание элементов АРМ экономиста подразделения обеспечили программе высокую надежность [3].

Телефон 264-73-41, Москва

ЛИТЕРАТУРА

1. Мартин Дж. Организация баз данных в вычислительных системах. — М.: Мир, 1980.
2. Волков Д. К. Стандартизация структур записей наборов данных // Сб. Вопросы судостроения, Сер. Математические методы, программирование, эксплуатация ЭВМ. — Л.: ЦНИИ «РУМБ». — 1979. — Вып. 17.
3. Липаев В. В., Потапов А. И. Длительность разработки сложных программных средств // Микропроцессорные средства и системы. — 1986. — № 6.

Статья поступила 01.09.87

УДК 681.3.06

С. П. Максимяк, Ю. Ю. Сорокин, В. В. Субач

НЕПОСРЕДСТВЕННЫЙ ДОСТУП К ДАННЫМ В ФАЙЛАХ НА ПАСКАЛЕ-1

Распространенная на микроЭВМ «Электроника 60» ДВК система программирования Паскаль-1 (ОС ДВК, РАФОС, ФОДОС) не поддерживает непосредственного доступа к данным в файлах*. Поэтому возможности эффективной реализации ряда алгоритмов манипуляции данными ограничиваются**. Использование в программах на Паскале-1 вызовов внешней процедуры ASEEK (рис. 1) позво-

* Операционная система СМ ЭВМ РАФОС. Документация системы, т. 5, ч. 6, кн. 1—3, 1983.

** Грогоно П. Программирование на языке Паскаль.— М.: Мир, 1982.

```
(*$E+*)
TYPE
  FILEDESCRIPTOR=RECORD
                   HIGHW,LOLD,ROLD,LNEW,RNEW,RECRD,FIL,
                   BNEED: INTEGER; ADRBUF: ^INTEGER;
                 END;
PROCEDURE ASEEK(VAR FD:FILEDESCRIPTOR);
  CONST BLKSIZ=256;
  VAR J: INTEGER;
  BEGIN
  (*$C
        .RADIX  8.
        .ENABL  LC,GBL
;
;       СМЕЩЕНИЯ В ПРОГРАММНОМ ДЕСКРИПТОРЕ ФАЙЛА
;
HIW$=   0       ;ДЛИНА ЗАПИСИ-1
L$=     2       ;ЛЕВАЯ ...
R$=     4       ;...И ПРАВАЯ ГРАНИЦЫ ЗАПИСИ В БЛОКАХ В СЛОВАХ
LL$=    6       ;ТО ЖЕ, НО ТРЕБУЕМЫЕ
RR$=    10
NR$=    12      ;НОМЕР ТРЕБУЕМОЙ ЗАПИСИ
FIL$=   14      ;БУФЕР УКАЗАТЕЛЯ НА ФАЙЛОВЫЙ ДЕСКРИПТОР PASCAL
NN$=    16      ;НОМЕР ТРЕБУЕМОГО БЛОКА
VBF$=   20      ;АДРЕС СВЯЗАННОЙ ПЕРЕМЕННОЙ(ФОРМИРУЕТСЯ ВНУТРИ)
;
;       СМЕЩЕНИЯ В БЛОКЕ ОПИСАНИЯ ФАЙЛА ИСП.СИСТЕМЫ PASCAL
;
S$A=    0       ;АДРЕС БУФЕРНОЙ ПЕРЕМЕННОЙ
C$N=    2       ;НОМЕР КАНАЛА
O$C=    3       ; И КОД ОПЕРАЦИИ ДЛЯ EMT
B$N=    4       ;НОМЕР ТЕКУЩЕГО БЛОКА
A$B=    6       ; И МЕСТО,ГДЕ СИДИТ ЕГО КОПИЯ В ОЗУ
S$F=    12      ;СЛОВО СТАТУСА ФАЙЛА
;
;       МАКРОКОМАНДА ДЛЯ ПЕРЕЗАПИСИ ПОЛЕЙ ДАННЫХ И ВСПОМОГАТЕЛЬ
                                              НЫХ ДЕЙСТВИЙ
;
.MACRO  COPY    A0,A1,A2,C1,C2,MODE
.MACRO  AT      A,OP
.IF   IDN A,#0
        CLR     OP
.IFF
.IF   DIF A,OP
        MOV     A,OP
.ENDC
.ENDC
.ENDM
        AT      A0,R0
        AT      A1,R1
        AT      A2,R2
.IIF NB <C1>,     C1
.IIF NB <C2>,     C2
        CALL    MODE'COP
.ENDM
        .RADIX  10.
*)
(* ВЫЧИСЛЕНИЕ ГРАНИЦ ЗАПИСИ И НОМЕРА ТРЕБУЕМОГО БЛОКА
   ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОВ ПАСКАЛЯ ДЕЛАЕТ ЭТУ ОПЕРАЦИЮ
   НЕЗАВИСИМОЙ ОТ ТИПА ПРОЦЕССОРА *)
IF FD.RECRD)=0 THEN BEGIN
        J:=FD.HIGHW+1;
        FD.RNEW:=((FD,RECRD+1)*J-1) MOD BLKSIZ;
        FD.LNEW:=(FD.RECRD*J) MOD BLKSIZ;
        FD.BNEED:=(FD.RECRD*J) DIV BLKSIZ
        END
      ELSE
        FD.BNEED:=-1;
(*$C
        .RADIX  8.
        BR      *ASEEK
;
;       ПОДПРОГРАММЫ КОПИРОВАНИЯ ОБЛАСТЕЙ ДАННЫХ(ДЛЯ РАБОТЫ COPY)
;
SBCOP:  MOV     VBF$(4),SRCB
        MOV     A$B(5),DSTB
        BR      STRO
BSCOP:  MOV     A$B(5),SRCB
        MOV     VBF$(4),DSTB
STRO:   ASL     R0
        ASL     R1
        INC     R1
        ASL     R2
CLOOP:  116062
SRCB:   0
DSTB:   0
        CMPB    (0)+,(2)+
        CCC
        CMP     R0,R1
        BLE     CLOOP

        RETURN
;
;       ПОДПРОГРАММЫ ЧТЕНИЯ-ЗАПИСИ БЛОКА
;
BWRI:   MOVB    #11,O$C(5)
        BR      EMTS
BREA:   MOVB    #10,O$C(5)
EMTS:   MOV     S$F(5),-(SP)
        CLR     S$F(5)
        MOV     R5,R0
        TST     (0)+
        EMT     375             ;ВЫЗОВ .READW/.WRITW
        BCS     ERRS
        MOV     (SP)+,S$F(5)
        RETURN
ERRS:   JSR     R5,RTERR        ;АВАРИЙНЫЙ ВЫХОД
        FATLIO
;
;       ОСНОВНОЙ МОДУЛЬ
;
$ASEEK::.IRP    X,<R0,R1,R2,R4,R5>
        MOV     X,-(SP)
        .ENDR
        MOV     16(SP),R4       ;УКАЗАТЕЛЬ НА ПРОГ.ДЕСКРИПТОР
        MOV     FIL$(4),R5      ;
        CMP     L$(4),R$(4)     ;В ОДНОМ ЛИ БЛОКЕ НАЧАЛО И КОНЕЦ?
        BEQ     THBLOC          ;ПЕРВАЯ ОПЕРАЦИЯ НА ДАННОМ КАНАЛЕ
        BGT     LSCAN           ;-->НЕТ!
SCAN:   COPY    #0,@R4,L$(4),,,SB
        BR      THBLOC
LSCAN:  COPY    @R4,R0,#0,<SUB  R$(4),R0>,,SB
        CALL    BWRI            ;=>НА ДИСК
        DEC     B$N(5)
        CALL    BREA            ;<=С ДИСКА ПРЕДЫДУЩИЙ БЛОК
        COPY    #0,@R4,L$(4),<SUB  R$(4),R1>,<DEC  R1>,SB
THBLOC: CMP     B$N(5),NN$(4)   ;ТОТ ЛИ БЛОК В БУФЕРЕ?
        BEQ     SCOPE           ;-->ТОТ
        CALL    BWRI            ;НЕТ--ВЫВЕСТИ БЛОК ИЗ БУФЕРА
        TST     NN$(4)          ;ЭТО ЗАКРЫТИЕ?
        BGE     OTHB            ;-->НЕТ:НОМЕР БЛОКА ПОЛОЖИТЕЛЕН
        MOV     R5,FILE
        MOV     #1,B$N(5)
        CALL    CLOSE+4
        BR      EXSEEK
OTHB:   MOV     NN$(4),B$N(5)   ;ЧИТАЕМ НУЖНЫЙ БЛОК
        CALL    BREA
SCOPE:  MOV     LL$(4),L$(4)    ;КОПИРОВАНИЕ НОВЫХ В СТАРЫЕ
        MOV     RR$(4),R$(4)
        CMP     L$(4),R$(4)     ;В ОДНОМ ЛИ БЛОКЕ...
        BGT     LSEE
SEE:    COPY    L$(4),R$(4),#0,,,BS
        BR      EXSEEK
LSEE:   COPY    L$(4),#377,#0,,,BS
        CALL    BWRI
        INC     B$N(5)
        CALL    BREA
        COPY    #0,R$(4),@R4,<SUB  R$(4),R2>,,BS
EXSEEK: .IRP    X,<R5,R4,R2,R1,R0>
        MOV     (SP)+,X
        .ENDR
        .RADIX  10.

*)
END;
```

Рис. 1. Исходный текст процедуры ASEEK

ляет устранить эти ограничения для работы с записями длиной до 256 слов.

Единственным параметром процедуры является ссылка на программный дескриптор файла, представляющий собой запись длиной в девять слов. Последнее слово дескриптора хранит адрес переменной (буфера файла), в которой содержится копия текущей записи. Седьмое слово формируется так, как показано в примере на рис. 2. Назначение остальных элементов дескриптора понятно из комментариев к исходному тексту ASEEK на языках Макро-11, Паскаль-1.

До первого использования процедуры позиционирования ASEEK необходимо программно инициализировать 1-...3-е и 6-...9-е слова дескриптора, а при последующих обращениях к процедуре указывать лишь номер требуемой записи (шестое слово). Записи нумеруются с нуля, указание отрицательного номера записи интерпретируется как закрытие файла. При попытке позиционирования буфера файла за пределы последнего выдается диагностическое сообщение исполняющей системы и выполнение программы прекращается.

Для закрытия файла ASEEK неявно использует стандартную процедуру исполняющей системы. Однако прямой вызов CLOSE (F) для закрытия файла, с которым работала ASEEK, запрещен.

Процедура ASEEK подразумевает размещение записей в файле с переходом через границу блока, что обес-

```
PROGRAM TSEEK;
TYPE
        (*ТИП ЗАПИСИ В ФАЙЛЕ ПРЯМОГО ДОСТУПА*)
        TESTRECORD=RECORD
                        N1,N2,N3: INTEGER;
                    END;
        FILESEEK=FILE OF TESTRECORD;
        (*ТИП ПРОГРАММНОГО ДЕСКРИПТОРА ФАЙЛА*)
        FILEDESCRIPTOR=RECORD
                        HIGHW,LOLD,ROLD,LNEW,RNEW,RECRD,FIL,
                        BNEED:INTEGER; ADRBUF:^TESTRECORD;
                       END;
VAR
        FD:FILEDESCRIPTOR;
        F:FILESEEK;
        (*ПЕРЕМЕННАЯ-БУФЕР(ХРАНИТ ТЕКУЩУЮ ЗАПИСЬ*)
        TESTSTRUCTURE:TESTRECORD;
        K:INTEGER;
PROCEDURE ASEEK(VAR D:FILEDESCRIPTOR);EXTERNAL;
BEGIN
 RESET(F,'DK:TSTFIL','DAT');
 (*C                            ;ИНИЦИАЛИЗАЦИЯ ПРОГРАММНОГО
                                  ДЕСКРИПТОРА
        MOV     FILE,R0         ;СВЯЗЫВАНИЕ С ОПИСАТЕЛЕМ ФАЙЛА
                                   ИСПОЛНЯЮЩЕЙ
        MOV     R0,K(R5)        ;СИСТЕМЫ
        DEC     4(R0)
*)

FD.FIL:=K;
FD.HIGHW:=2;
FD.LOLD:=0;
FD.ROLD:=0;
FD.ADRBUF:=@TESTSTRUCTURE;
REPEAT
        WRITE('POS #:');READLN(FD.RECRD);
        ASEEK(FD);
        IF FD.RECRD>=0 THEN BEGIN
        WRITELN('OLD:',TESTSTRUCTURE.N1:7,
                       TESTSTRUCTURE.N2:7,
                       TESTSTRUCTURE.N3:7);
        WRITE('NEW:');
        READLN(TESTSTRUCTURE.N1,TESTSTRUCTURE.N2,TESTSTRUCTURE.N3);
        END;
UNTIL
        FD.RECRD<0
END.
```

Рис. 2. Пример использования процедуры ASEEK: чтение и модификация записей из трех целых чисел в файле DK:TSTFIL. DAT

печивает эффективное использование дискового пространства. Это обстоятельство необходимо учитывать при работе по прямому доступу с файлами, сформированными стандартными WRITE (F,...) и PUT (F).

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

Сообщение поступило 29.10.87

Дополнение к статье Гостева А. В. «Ускорение считывания с диска в операционной системе CP/M-80» (№ 6, 1987 г.)

В связи с тем, что у некоторых программистов возникли затруднения при использовании программы TURBO на микроЭВМ СМ1800, довожу до их сведения следующее.

Программа TURBO эффективно работает на моделях СМ1800, имеющих монитор системы «MONID 1.0».

На более поздних моделях СМ1800, имеющих монитор «MONID 1.3», использовать программу TURBO нецелесообразно вследствие существенно меньшего быстродействия контроллера дисков.

Программа TURBO могла бы столь же успешно функционировать и на новых моделях СМ1800, если бы новые модели не оказались хуже старых по быстродействию обмена с дисками.

Автор приносит свои извинения пользователям новых моделей СМ1800 и просит учесть, что, не будучи разработчиком СМ1800, три года назад не имел никаких сведений об их существовании и поэтому не отразил это обстоятельство в статье.

Телефон 333-78-73, Москва

САПР ЦИФРОВЫХ УСТРОЙСТВ НА ПЛИС

ПЛИС — ИНСТРУМЕНТ РАЗРАБОТЧИКА МП-СИСТЕМ

Большинство современных специализированных БИС (СП БИС) для построения высокоэффективных микропроцессорных систем проектируется на основе полузаказных БИС — базовых матричных кристаллах (БМК) и программируемых лигических интегральных схемах (ПЛИС). Выбор исходной полузаказной БИС зависит от стадии жизненного цикла системы: разработки и подготовки к серийному производству, серийного освоения (роста), внедрения (зрелости) и старения (спада). На этапах разработки и роста системы целесообразно использовать недорогие, доступные в изготовлении за короткий срок и легко заменяемые ПЛИСы. На этапе зрелости в условиях больших объемов производства предпочтительнее БМК с высокой плотностью упаковки.

Таким образом, при проектировании системы будут учтены не только технические характеристики БИС, но и экономические аспекты их применения, что значительно снижает общую стоимость системы.

Для изделий, выпускаемых малыми и средними тиражами (до 2000 шт.), применение ПЛИС в качестве основы для СП БИС в два раза выгоднее по сравнению с ИС малой и средней степени интеграции и в три-четыре раза выгоднее, чем использование БМК и полностью заказных БИС. Особенно существенный выигрыш ПЛИС дают в сроках разработки и готовности изделия, так как они обеспечивают 5-...10-кратное ускорение проектирования СП БИС по сравнению с другими технологиями*.

При этом важнейшими факторами, гарантирующими успех, являются следующие преимущества технологии ПЛИС: сжатые сроки поставки готовых качественных ПЛИС; широкий ассортимент и разнообразные технические возможности; поддержка средств конструкторской разработки с помощью ЭВМ; простота перехода от ПЛИС к БМК и полностью заказным БИС; состояние и обнадеживающие перспективы развития номенклатуры отечественных ПЛИС.

На мировом рынке специализированных интегральных схем наблюдается значительное укрепление позиций ПЛИС. Первый ПЛИС 82S100 и PAL16 выпущены фирмами Signetics и MMI. В настоящее время разработкой и внедрением этих микросхем заняты фирмы AMD, Fairchild Semiconductor, Harris Semiconductor, Intel, GE/Intersil, Altera, Lattice Semiconductor, Uesi Technology, Xilinx, Motorola. Прогнозируемый объем продаж ПЛИС к 1990 г. составит 1,02 млрд. долл. при общем объеме продаж специализированных ИС — 13,7 млрд. долл.**.

Отечественная промышленность предлагает разработчикам систем ПЛИС двух типов: программируемые логические матрицы (ПЛМ) и микросхемы программируемой матричной логики (ПМЛ). К первым относятся микросхемы серий К556 (РТ1...РТ3), ко вторым — серии КР1556 (ХЛ8, ХП4, ХП6, ХП8). ПЛИС серии К556 содержат программируемые матрицы И и ИЛИ, а ПЛИС серии КР1556 — только матрицы И.

В публикуемой тематической подборке статей приводятся общие сведения о САПР цифровых устройств на ПЛИС. Первые три статьи содержат технические характеристики, описание структуры, режимов языка программирования и системы проектирования ПМЛ серии КР1556. В остальных двух статьях даны обобщенные характеристики системы проектирования устройств на основе ПЛМ серии К556.

---

* Microprocessors and microsystems.— 1988.— Vol. 12, № 5.— P. 260—263.

** Электроника.— 1986.— № 2. С. 41—53.

УДК 681.326.3

И. В. Красильников, П. С. Приходько, В. М. Чурков, Ю. И. Щетинин

МИКРОСХЕМЫ ПРОГРАММИРУЕМОЙ МАТРИЧНОЙ ЛОГИКИ СЕРИИ КР1556

Микросхемы программируемой матричной логики (ПМЛ) представляют собой многовходовые структуры И-ИЛИ и И-ИЛИ-ТГ с возможностью организации межэлементных обратных связей и формирования входных конъюнкций посредством программируемых полей матриц И.

В состав серии КР1556 входят следующие БИС:

КР1556ХЛ8 — многофункциональная программируемая логическая матрица с обратными связями, реализующая восемь выходных функций от 16 входных переменных, включающих до 56 термов.

КР1556ХП4, КР1556ХП6, КР1556 ХП8 — многофункциональные программируемые логические матрицы с обратными связями, реализующие восемь выходных функций от 16 переменных; четырьмя (шестью, восьмью) D-триггерными трехстабильными выходами с внешней синхронизацией.

Микросхема КР1556ХЛ8 (рис. 1).

В программируемой матрице И (64 строки, 32 столбца) образуются конъюнкции входных переменных A0...A9, причем каждая входная переменная может иметь прямое или инверсное значение, или не входить в конъюнкцию совсем. Это первый уровень логики прибора.

Выходные сигналы, появляющиеся на выходных шинах матрицы И, вводятся в матрицу ИЛИ, образующую второй логический уровень и реализующую дизъюнкции заданных конъюнкций.

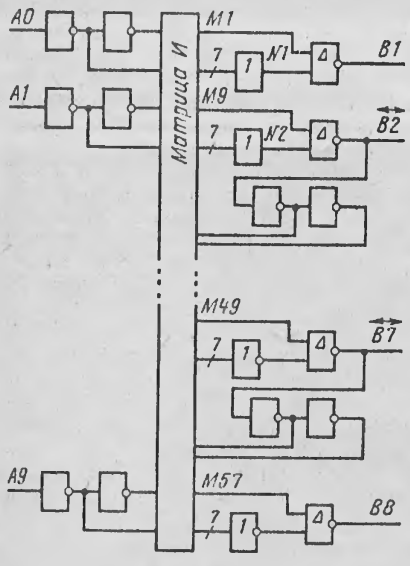


Рис. 1. Структурная схема микросхемы КР1556ХЛ8

Матрицу ИЛИ образуют восемь дизъюнкторов, каждый из которых может быть связан с одной из восьми групп конъюнкторов (по семь конъюнкторов в группе).

Выходной каскад, включающий восемь выходных усилителей с тремя состояниями, предназначен для вывода информации. Выборка выходных усилителей осуществляется непосредственно из матрицы И, т. е. адресным путем. Возможны следующие режимы работы микросхемы: хранения, считывания, программирования и контроля матрицы И.

Режим хранения осуществляется установкой выходов схемы в закрытое (высокоимпедансное) состояние путем подачи на выводы определенных адресных воздействий.

В режиме считывания при подаче кода адреса на входы запрограммированной микросхемы адресные формирователи подают соответствующие уровни на столбцы матрицы И. Одновременно в матрицу через адресные формирователи обратной связи поступает выходной сигнал. В зависимости от карты пережигания перемычек на выходе схемы реализуется логическая функция входных и выходных сигналов.

Разрешение выборки выходных каскадов осуществляется непосредственно из матрицы. Пережигание всех перемычек в управляющей строке соответствует переводу выходного каскада в режим считывания независимо от подаваемого адреса.

Двунаправленные выходы микросхем могут быть использованы как входы, если установить выходной каскад в третье состояние. Работа микросхемы КР1556ХЛ8 поясняется табл. 1. Условное графическое обозначение КР1556 ХЛ8 приведено на рис. 2; назначение выводов — в табл. 2.

Первый уровень логики микросхемы

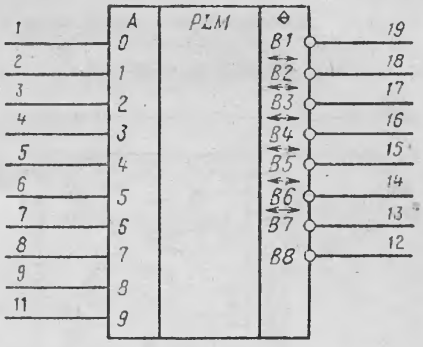


Рис. 2. Условное графическое обозначение микросхемы КР1556ХЛ8

Таблица 2

Назначение выводов микросхемы КР1556ХЛ8

Вывод	Назначение
1...9, 11	Адресные входы, A0...A9
12, 19	Выходы, B8, B1
13...18	Входы-выходы, B7...B2
10	Общий
20	Напряжение питания, U_{cc}, 5 В

КР1556ХП4 (рис. 3) образуют конъюнкции входных переменных A0...A7; второй — матрица ИЛИ, состоящая из восьми дизъюнкторов (четырех 7-входовых и четырех 8-входовых).

Выходной каскад КР1556ХП4 включает четыре выходных усилителя с тремя состояниями, предназначенных для ввода-вывода информации (аналогично КР1556ХЛ8), и четыре D-триггера, синхронизируемых фронтом внешнего сигнала C, причем выходы триггеров подключены к матрице И и контактам микросхем через трехстабильные буферы, управляемые внешним сигналом OE.

Логические состояния микросхемы КР1556ХП4 (триггерные выходы) определяются из табл. 3. Условное графи-

Таблица 1

Таблица функционирования микросхемы КР1556ХЛ8

Выход терма M_i, i=1,9...57	Выход матрицы ИЛИ N_j, j=1,2...8	Выход B_j, j=1,2...8	Режим работы
L	L	Z	Хранение
L	H	Z	»
H	L	H	Считывание
H	H	L	»

Примечание. N — сумма конъюкций входных A0...A9 и выходных B2...B7 переменных.

ческое изображение микросхемы КР1556ХП4 представлено на рис. 4, назначение выводов — в табл. 4.

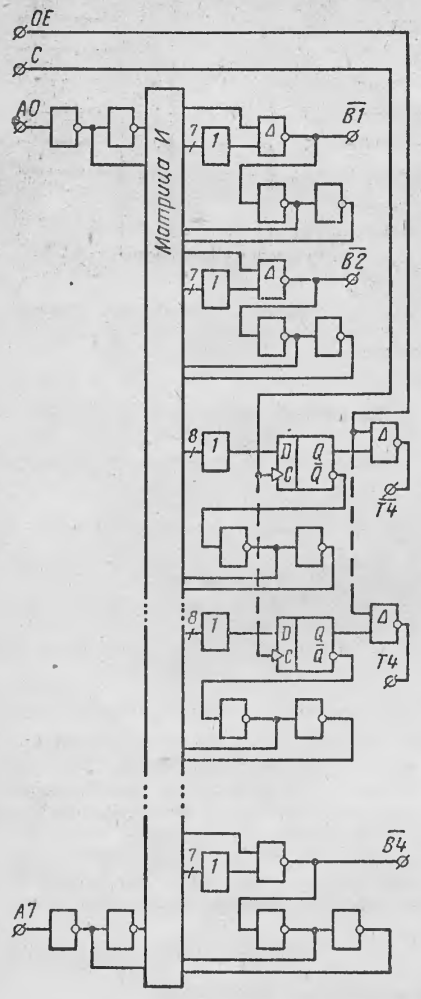


Рис. 3. Структурная схема микросхемы КР1556ХП4

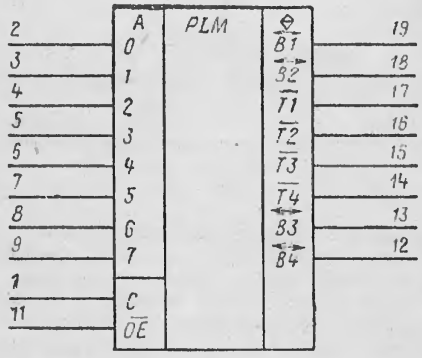


Рис. 4. Условное графическое обозначение микросхемы КР1556ХП4

Таблица 3

Таблица функционирования микросхемы КР1556ХП4

OE	C	Выход матрицы ИЛИ N_i, i=3, 4, 5, 6	Триггерные выходы Q_t	Q_{t+1}	Выход T_j, j=1, 2, 3, 4	Режим работы
L	X	L	L	L	H	Считывание
L	X	L	H	H	L	»
L	X	H	L	L	H	»
L	X	H	H	H	L	»
L	↓	L	L	L	H	»
L	↓	L	H	L	H	»
L	↓	H	L	H	L	»
L	↓	H	H	H	L	»
H	X	X	X	X	Z	Хранение
H	|	X	X	X	Z	»

Микросхемы КР1556ХП6 и КР1556 ХП8 различаются числом комбинационных и триггерных выходов: КР1556ХП6 имеет шесть триггерных и два комбинационных выхода; КР1556 ХП8 — восемь триггерных выходов. Потребляемый ток — не более 200 мА; время задержки распространения сигнала — не более 40 нс; температура окружающей среды —10...+70° С.

Режимы программирования микросхем серии КР1556 однотипны и осуществляются по следующей методике.

Программирование поля матрицы И происходит при повышенном до уровня 11,5+0,5 В напряжении питания в два этапа.

На этапе 1 программируются строки 0...31, на втором — 32...63. На выводы 1 и 11 (для этапов 1 и 2 соответственно) также подается напряжение U=11,5± ±0,5 В.

Для пережигания отдельной перемычки необходимо выбрать соответствующие столбец и строку матрицы. Требуемый столбец (подключение выводов микросхем КР1556ХП8, КР1556ХП4 в матрицу И показано в табл. 5, 6) выбирается подключением

Таблица 4

Назначение выводов микросхемы КР1556ХП4

Вывод	Назначение
1	Синхронизация, С
2...9	Адресные входы, А0...А7
11	Разрешение выхода, OE
14...17	Выходы, Т1...Т4
12, 13, 18, 19	Входы-выходы, В1...В4
10	Общий
20	Напряжение питания U_{cc} 5 В

Таблица 5

Обозначение столбцов матрицы микросхем КР1556ХЛ8 и КР1556ХП4 (0...15)

Микросхема	Столбец															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
КР1556ХЛ8	A0	$\overline{A0}$	B1	$\overline{B1}$	A1	$\overline{A1}$	B2	$\overline{B2}$	A2	$\overline{A2}$	T1	$\overline{T1}$	A3	$\overline{A3}$	T2	$\overline{T2}$
КР1556ХП4	A0	$\overline{A0}$	B1	$\overline{B1}$	A2	$\overline{A2}$	B2	$\overline{B2}$	A3	$\overline{A3}$	B3	$\overline{B3}$	A4	$\overline{A4}$	B4	$\overline{B4}$

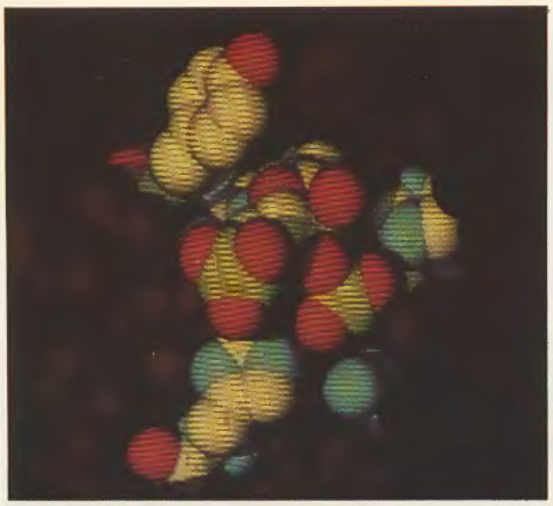
Таблица 6

Обозначение столбцов матрицы микросхем КР1556ХЛ8 и КР1556ХП4 (16...31)

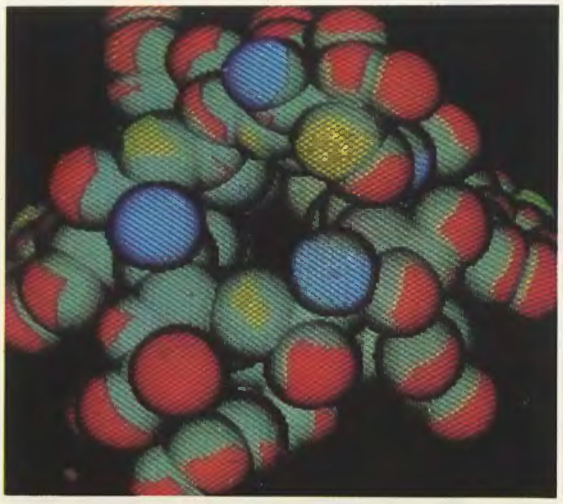
Микросхема	Столбец															
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
КР1556ХЛ8	A4	$\overline{A4}$	T3	$\overline{T3}$	A5	$\overline{A5}$	T4	$\overline{T4}$	A6	$\overline{A6}$	B3	$\overline{B3}$	A7	$\overline{A7}$	B4	$\overline{B4}$
КР1556ХП4	A5	$\overline{A5}$	B5	$\overline{B5}$	A6	$\overline{A6}$	B6	$\overline{B6}$	A7	$\overline{A7}$	B7	$\overline{B7}$	A8	$\overline{A8}$	A9	$\overline{A9}$

ДИАЛОГОВАЯ ГРАФИЧЕСКАЯ СИСТЕМА ДЛЯ ИССЛЕДОВАНИЯ СТРУКТУР СЛОЖНЫХ МОЛЕКУЛЯРНЫХ ОБЪЕКТОВ МОЛЛЮСК-2

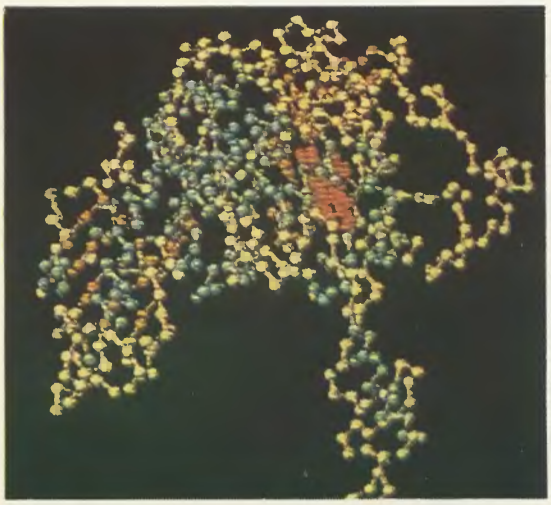
(К ст. Бобкова В. А., Кислюка О. С., Боровиной Т. А.)



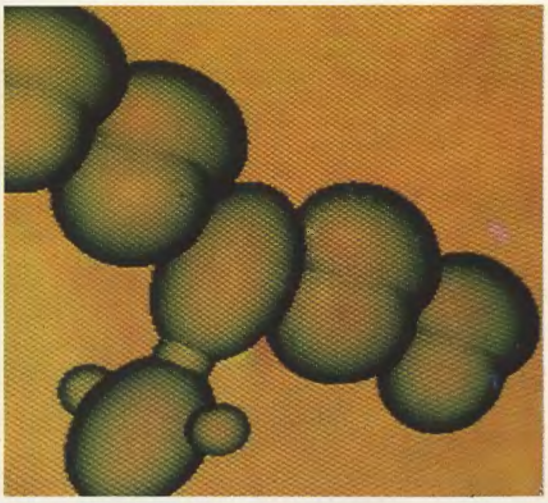
Ван-дер-Ваальсова модель, содержащая более трех тысяч атомов



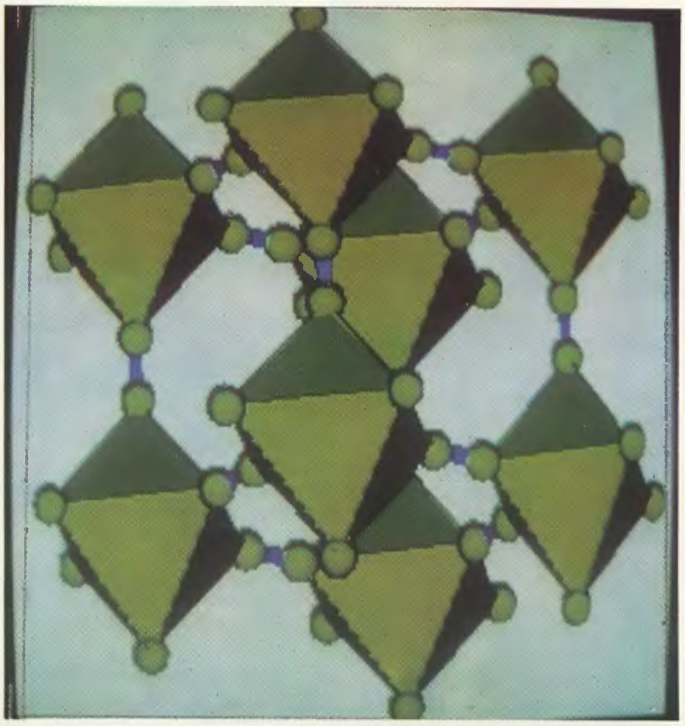
Ван-дер-Ваальсова поверхность «обкатанная» шаром, моделирующим молекулу растворителя. Участки, недоступные растворителю, окрашены в серый цвет



Шаро-стержневая модель



Модель молекулярных орбиталей, изображенных в виде набора эликсоидов



Модель кристаллической решетки в виде совокупности выпуклых многогранников

(Продолжение на 4 с. вкладки)

Вы хотите повысить культуру производства, оперативность и эффективность управления, сократить аппарат и подсобные службы и получить ощутимый коммерческий успех..
СП ИНТЕРМИКРО поможет Вам в этом.

PROSYSTEM
Computer Technic

С широким развитием и внедрением вычислительной техники автоматизация проникла и в сферы издательской и типографской деятельности. Как и в приложении к другим сферам, вычислительная техника в издательствах дает возможность резко повысить производительность труда сотрудников, избавиться от многих рутинных ручных операций , сэкономить средства и значительно улучшить временные и качественные характеристики технологиче ских процессов.

НАСТОЛЬНАЯ ТИПОГРАФИЯ (DESKTOP PUBLISHING SYSTEM) это комплект оборудования, предназначенного для издания небольших тиражей различной документации с типографским качеством и изготовления оригинал-макета с последующей печатью больших тиражей в типографии. В комплект настольной типографии может входить полиграфическое оборудование с возможностью печати до 10 тысяч листов в час.



НАСТОЛЬНАЯ ТИПОГРАФИЯ включает в себя:

персональный компьютер - обычно PC AT с операционной памятью 1 Мбайт и специальными аппаратными средствами для связи с лазерным принтером и сканнером и программой автоматизированной верстки;

сканнер - устройство для ввода в персональный компьютер текста и изображений (300x300 или 600x600);

лазерный принтер - для получения небольших тиражей и оригинал-макета издания (300x300 или 600x600).

Математическое обеспечение настольной типографии содержит широкий ассортимент общепринятых и декоративных гарнитур и учитывает все полиграфические нормы, принятые в СССР.

Процесс изготовления печатной формы в настольной типографии начинается с ввода текста с клавиатуры или со сканнера. Специальные программы позволяют произвести начальную обработку текста: расстановление переносов и проверку правописания. Откорректированный текст верстается вместе с фотографиями или рисунками, введенными со сканнера, и на лазерном принтере Вы получаете оригинал-макет или небольшой тираж (200-300 экземпляров). Если тираж большой, то оригинал-макет может быть передан для дальнейшей обработки в типографию. Возможно также соединить настольную типографию с фотонаборным устройством для получения готовой фотоформы.

Настольная типография позволит Вам самим подготовить и издать любую типографскую продукцию - от газеты до пригласительного билета и визитной карточки в черно-белом изображении. Оригинальные шрифты, разнообразное художественно-графическое оформление, высокое качество и скорость исполнения при любом тираже печатной продукции делают НАСТОЛЬНУЮ ТИПОГРАФИЮ незаменимым помощником в работе.

НАСТОЛЬНАЯ ТИПОГРАФИЯ - ЭТО :

- Типографское качество изданий
- Возможность быстрого макетирования страниц, включающих как текстовые, так и графические материалы
- Широкий набор стандартных типографских шрифтов
- Возможность вставки в страницу изображений, полученных на сканнере
- Автоматическое обтекание изображения текстом
- Программы обработки русских и иностранных текстов, включающие автоматический перенос и проверку правописания
- Возможность автоматического считывания напечатанных, например, на пишущей машинке, русских и иностранных текстов
- Сложный набор (математические формулы, сноски, подписи к рисункам и таблицам)
- Простота и удобство создания таблиц
- Возможность работать в любых удобных для Вас единицах - сантиметрах, дюймах, пунктах
- Автоматический типографский контроль (число переносов в колонке, установка интерлиньяжа, удаление висячих строк)
- Автоматическое создание оглавления и индексных ссылок
- Автоматический контроль отбивки абзацев
- Ускорение издательских процессов
- Сокращение времени верстки
- Повышение качества макета
- Сокращение потребления бумаги
- Совмещение профессий
- Удешевление издания

Макет подготовлен на оборудовании фирмы PROSYSTEM группой Desktop Publishing, ИНТЕРМИКРО

СП ИНТЕР МИКРО

Настольная типография необходима Вам, если Вы работаете в: издательстве, научно - исследовательском институте, конструкторском или патентном бюро, министерстве, кооперативе, на заводе и в любом другом учреждении, где требуется оперативная и высококачественная полиграфия.

Мы поможем Вам создать законченную технологическую линию подготовки печатных документов и форм, рекламно-информационных проспектов, методических и регламентирующих брошюр и других необходимых Вашей фирме изданий.

Секретариат, библиотека, планово-финансовый отдел, патентное бюро и отдел кадров, оборудованные рабочими станциями на основе PC XT, смогут существенно улучшить свою работу, автоматизировав процесс хранения и выпуска документации, составления финансовых и договорных документов. Оборудование настольной типографии может легко включаться в компьютерную сеть вашего предприятия.



Включение настольной типографии в сеть предприятия, содержащую систему автоматизированного проектирования (САПР), даст возможность профессиональному конструктору сконцентрировать свои усилия на самом главном - на качестве самого проекта. Подготовка же конструкторской документации потребует меньшей квалификации, и при наличии готовых форм (форматка, штамп и т.д.), займет значительно меньше времени.

Если Ваша работа связана с обработкой большого количества документов и данных, то в компьютерную сеть, обслуживающую вашу фирму и включающую банк данных, настольная типография войдет как необходимое звено. Входящая документация будет вводиться с клавиатуры или со сканнера и направляться в банк данных, который может ее распределять по сети и, по мере надобности, выводить потребителям. Матричный принтер обеспечит достаточное для текущей документации качество, а в случае необходимости типографского качества документ через сетевое оборудование может быть передан в настольную типографию.Исходящая документация, поступившая в настольную типографию, преобразуется с использованием заготовленных форм, учитывающих требования к оформлению документов, и с типографским качеством поступает на размножение и в рассылку.

Специалисты совместного предприятия "Интермикро" осуществляют ввод в эксплуатацию и наладку оборудования, обучение пользователей и производят гарантийное и послегарантийное обслуживание поставляемой техники.

ОБРАЩАЙТЕСЬ:

ПРОСИСТЕМ

Вычислительная техника

Австрия
Austria, Parkring 12A/8/5,
A-1010, Vienna
Tel. 513-47-60
Tlx. 112937
Fax. 513-47-64

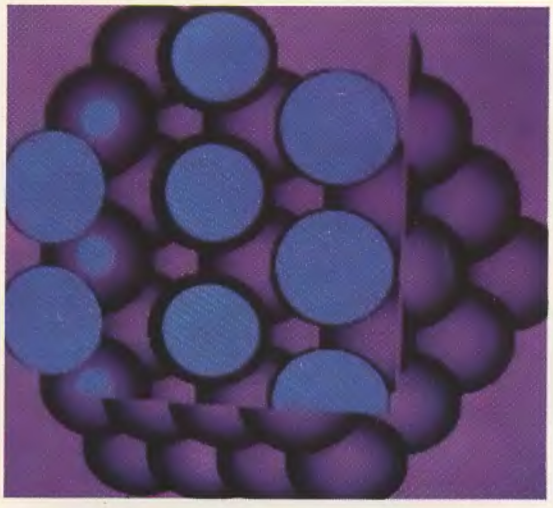
СССР
г.Москва, 123022,
ул. Трехгорный вал, д.2, стр.2
Тел. 255-01-01
Телекс 411941
Факс 230-22-92

ИНТЕРМИКРО

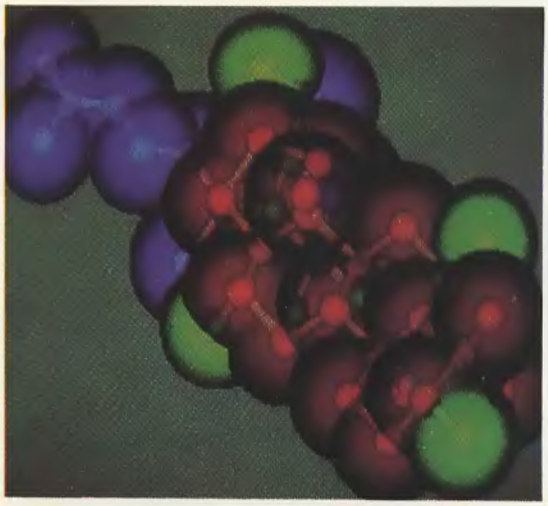
Совместное предприятие

СССР
г.Москва, 111024
ул.Авиамоторная, д.25/6
Тел. 253-96-86
Телекс 411035

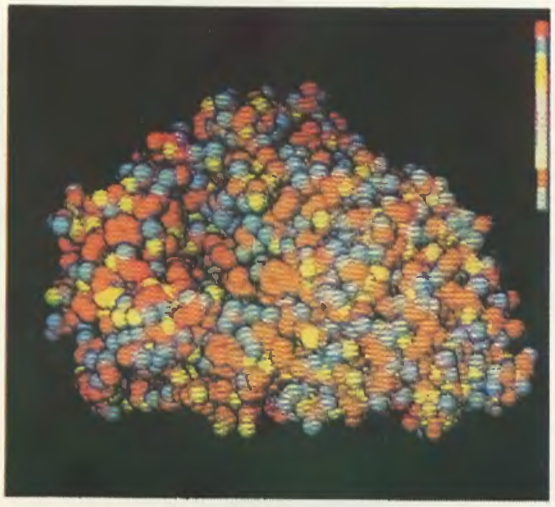
(Окончание. Начало на 1 с. вкладки)



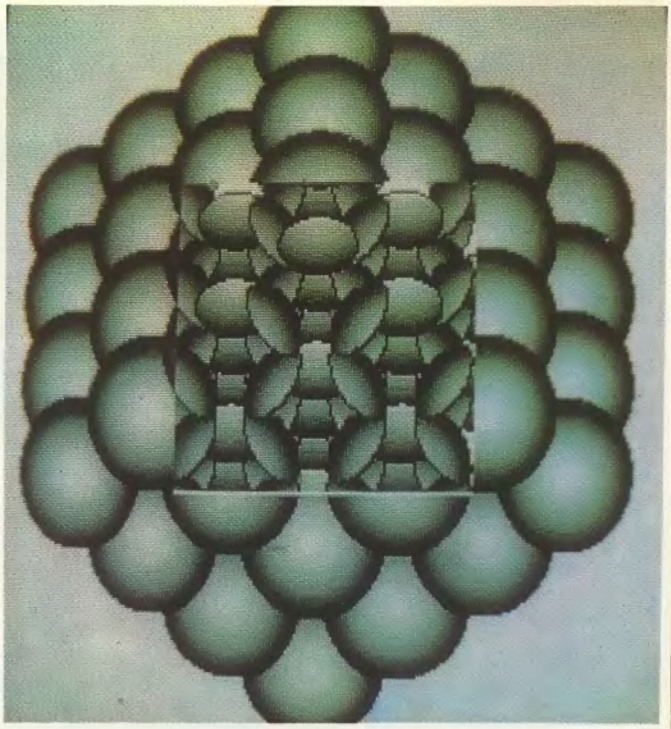
Изображение сечения объекта, состоящего из 64 шаров



Ван-дер-Ваальсова модель (крупно), освещенная вторым источником света



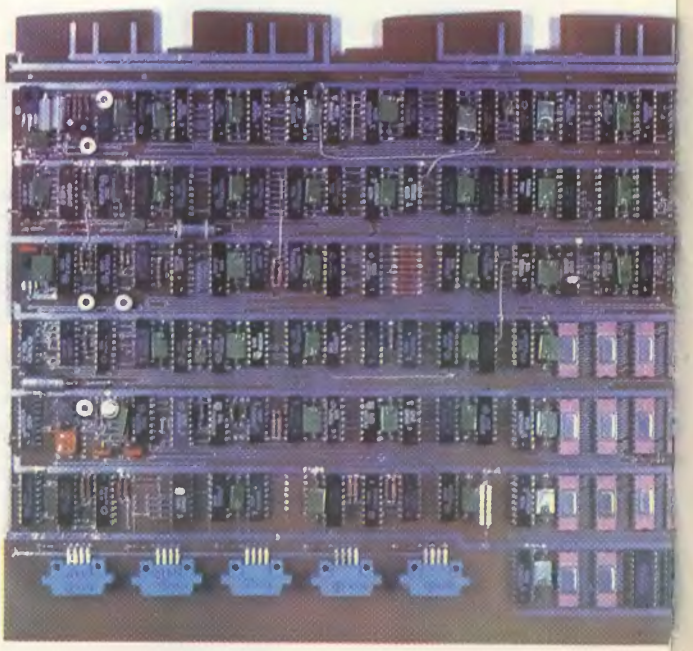
Полупрозрачное наложение Ван-дер-Ваальсовой и шаро-стержневой моделей



Объект, состоящий из 64 шаров. В середине экрана в «окне» изображены взаимные пересечения шаров, из которых состоит объект

БИНАРНАЯ СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ ДЛЯ ПРОМЫШЛЕННОГО ПРИМЕНЕНИЯ

(К ст. Матвеенко В. И., Староверова Ю. Г.)



Внешний вид платы устройства ввода видеоинформации

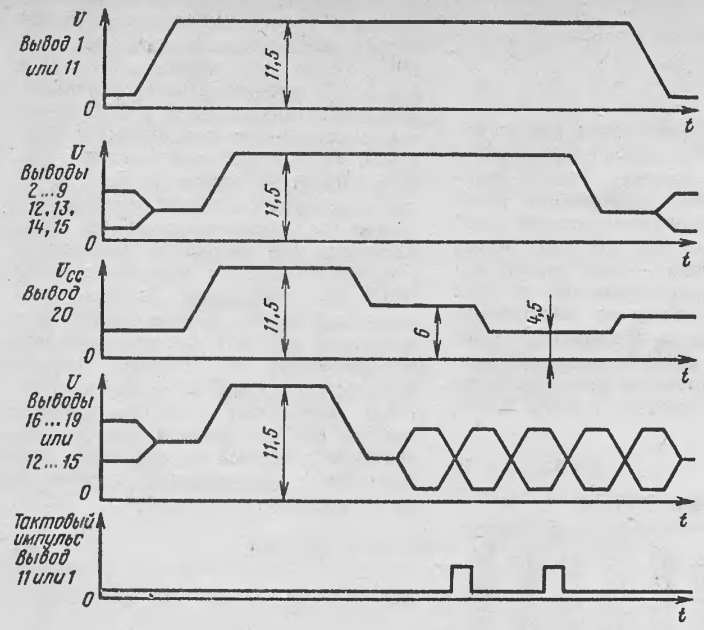


Рис. 5. Временная диаграмма процесса программирования микросхем серии КР1556

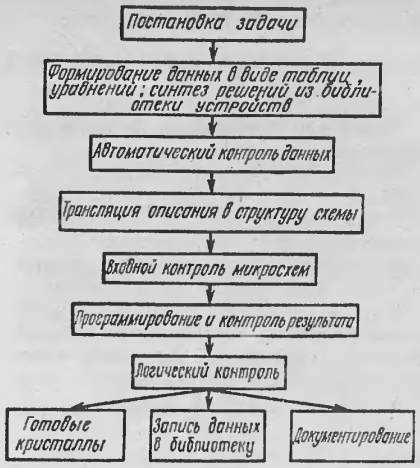


Рис. 6. Схема этапов проектирования и изготовления устройств на ПМЛ

Таблица 7

Таблица дешифрации столбцов микросхем КР1556

Столбец	Состояние вывода							
	9	8	7	6	5	4	3	2
0	РН	РН	РН	РН	РН	РН	РН	L
1	РН	РН	РН	РН	РН	РН	РН	H
2	РН	РН	РН	РН	РН	РН	РН	L
3	РН	РН	РН	РН	РН	РН	РН	H
4	РН	РН	РН	РН	РН	РН	L	РН
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
31	H	РН	РН	РН	РН	РН	РН	РН

Примечание. РН — программирующее напряжение U=11,5±0,5 В; L и H — состояния низкого и высокого уровней.

Таблица 8

Таблица дешифрации строк микросхем КР1556

Строки	Состояние вывода							
	12(19)	13(16)	14(17)	15(18)	16(12)	17(13)	18(14)	19(15)
0(32)	Z	Z	Z	Z	Z	Z	Z	РН
1(33)	Z	Z	Z	РН	Z	Z	Z	РН
2(34)	РН	H	РН	Z	Z	Z	Z	РН
3(35)	РН	H	РН	РН	Z	Z	Z	РН
4(36)	Z	РН	Z	Z	Z	Z	Z	РН
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
31(63)	РН	РН	РН	РН	РН	Z	Z	Z

Примечание. Z — состояние высокого импеданса (R=10 кОм).

выводов 2...9 в соответствии с табл. 7 и выводов 12...15 и 16...19 — в соответствии с табл. 8. Пережигание выбранной перемычки происходит при подаче программирующего импульса на один из выводов: 16...19 — для верхней и 12...15 — для нижней половин матрицы.

Понижая напряжение питания до 6 В и подавая тактовый импульс, контролируют отсутствие перемычки по выводам 16...19 для строк 0...31 и по выводам 12...15 для строк 32...63. При отсутствии перемычки на выходе должно быть напряжение низкого уровня $U_L \leqslant 0{,}5$ В.

Контроль на отсутствие перемычки повторяется при напряжении питания $U_{CC} = 4{,}5$ В. При наличии перемычки процедуру программирования следует повторять до 10 раз.

Временная диаграмма программирования микросхем серии КР1556 приведена на рис. 5.

Для создания устройств на БИС серии КР1556 разработана система проектирования, включающая в себя этапы от постановки задачи до получения готовых микросхем, реализующих заданные функции (рис. 6).

Система проектирования выполнена на базе вычислительного комплекса ДВК2 и имеет в своем составе развитое обеспечение для проектирования и изготовления устройств на рассмотренных ПМЛ.

Телефон 468-81-75, Москва

Статья поступила 15.07.88

УДК 681.326.3

В. М. Чурков, И. В. Красильников

ПОДСИСТЕМА ПРОЕКТИРОВАНИЯ НА ПМЛ

Меню для подсистемы, функции управления

Подсистема проектирования устройств на ПМЛ КР1556ХЛ8, КР1556ХП4, КР1556ХП6, КР1556ХП8 имеет меню следующего состава: «монитор», «набор данных», «набор данных в виде уравнений», «ввод данных с эталона», «входной контроль», «программирование», «сравнение с таблицей данных», «логический контроль», «распечатка данных», «запись файла», «чтение файла», «выход в систему».

«Монитор» предназначен для распечатки меню подсистемы и сокращенного наименования команд; «набор данных» — для набора информации, необходимой для программирования БИС ПМЛ с клавиатуры дисплея; «ввод данных с эталона» — для записи информации программирования с БИС ПМЛ, установленных на соединительной розетке модуля подсистемы в ОЗУ управляющей ЭВМ; «входной контроль» — для проверки целостности пережигаемых перемычек в БИС ПМЛ, установленных на розетке модуля подсистемы; «программирование» обеспечивает режим записи информации в БИС ПМЛ; «сравнение с таблицей данных» — сравнение данных программирования, находящихся в ОЗУ системы, с состоянием перемычек в БИС ПМЛ; «логический контроль» индицирует логические состояния БИС ПМЛ при задаваемых входных комбинациях переменных; «распечатка таблиц» предназначена для вывода на печатающее устройство таблицы истинности БИС ПМЛ или уравнений; «запись файла» — для записи файлов данных или уравнений на ГМД для последующего использования в команде «чтение файла»; «чтение файла» — для чтения с ГМД записанных ранее по команде «запись файла» файлов данных или уравнений; «выход в систему» — для перехода к операционной системе, в среде которой функционирует САПР.

Таблица 1

Обобщенная таблица истинности для ПМЛ серии КР1556

	КР1556ХЛ8	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	B1	B2	B3	B4	B5	B6
	КР1556ХП4	A0	A1	A2	A3	A4	A5	A6	A7	B0	B1	T2	T3	T4	T5	B6	B7
	КР1556ХП6	A0	A1	A2	A3	A4	A5	A6	A7	B0	T4	T2	T3	T4	T5	T6	B7
	КР1556ХП8	A0	A1	A2	A3	A4	A5	A6	A7	T0	T1	T2	T3	T4	T5	T6	T7
	0	1	1	0	X	1	0	1	X	X	X	1	0	X	X	0	X
	1	0	1	1	X	0	X	1	0	1	0	0	1	1	0	0	X
B0	2	X	X	0	0	1	1	X	X	1	0	1	0	1	X	1	0
(T0)	3	X	X	1	X	X	0	1	1	0	1	0	1	X	X	0	1
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	8	0	X	0	X	X	X	0	1	X	0	1	X	X	X	1	0
	9	X	0	0	0	1	1	1	1	0	0	0	0	X	X	X	X
	10	1	0	1	0	X	X	X	X	X	X	X	1	X	X	1	0
	11	1	1	1	1	X	X	X	X	0	0	0	0	X	X	X	X
B1	12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
(T1)	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	. . . . .																
	57	X	1	0	0	X	X	1	0	0	0	1	1	1	0	1	X
	58	X	X	X	X	0	0	X	0	X	X	X	1	X	X	0	1
	59	0	X	X	0	1	1	0	X	X	0	X	X	X	X	X	X
B7	60	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
(T7)	61	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Таблица 2

Обозначение выводов микросхем ПМЛ

Микросхема	Контакт																	
	1	2	3	4	5	6	7	8	9	11	12	13	14	15	16	17	18	19
КР1556ХЛ8	A8	A0	A1	A2	A3	A4	A5	A6	A7	A9	$\overline{B7}$	$\overline{B6}$	$\overline{B5}$	$\overline{B4}$	$\overline{B3}$	$\overline{B2}$	$\overline{B1}$	$\overline{B0}$
КР1556ХП4	C	A0	A1	A2	A3	A4	A5	A6	A7	ВК	B7	B6	T5	T4	T3	T2	B1	B9
КР1556ХП6	C	A0	A1	A2	A3	A4	A5	A6	A7	ВК	B7	T6	T5	T4	T3	T2	T1	B0
КР1556ХП8	C	A0	A1	A2	A3	A4	A5	A6	A7	ВК	T7	T6	T5	T4	T3	T2	T1	T0

Входной язык

Подготовка данных для программирования БИС ПМЛ проводится с использованием языка структурных таблиц (табл. 1) и систем булевых уравнений.

Данные в таблице могут находиться в следующих состояниях:

1. В конъюнкцию включается прямое значение переменной, при этом в соответствующий разряд необходимо записать 1.

2. В конъюнкцию включается инверсное значение переменной и записывается 0.

3. Переменная в конъюнкцию не включается, набирается символ X.

4. Если переменная не запрограммирована, то на экране отображается символ «—». При этом данная конъюнкция не включается в выходную функцию.

Переменные обозначаются тремя буквами: A, B, T. Переменные A0...A9 — входы; B0...B7 — входы-выходы; T0... T7 — выходы.

Представление данных в виде наглядных таблиц довольно удобно, однако для проектировщика в ряде случаев предпочтительнее булевые уравнения.

При выполнении режима «набор данных в виде уравнений» на экране распечатывается таблица, например для КР1556ХЛ8:

```
A8  A0  A1  A2  A3  A4  A5  A6
A7  3
A9 /B7 /B6 /B5 /B4 /B3 /B2 /B1
/B0 НП,
```

где A8 A0...3 соответствуют контактам 1...10 микросхемы; A9 / B7...НП — контактам 11...20.

Идентификация контактов БИС ПМЛ их условным обозначениям показана в табл. 2. Для схемы КР1556ХП6 таблица истинности, выводимая на экран дисплея, показана в табл. 3.

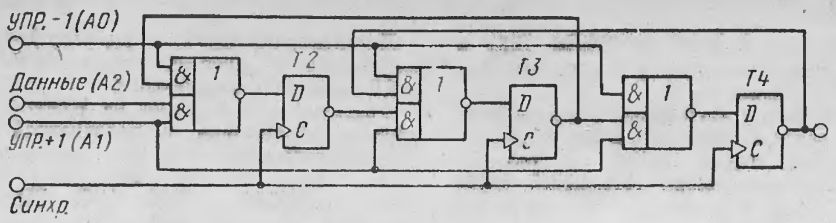


Рис. 1. Схема реверсивного сдвигающего регистра на D-триггерах

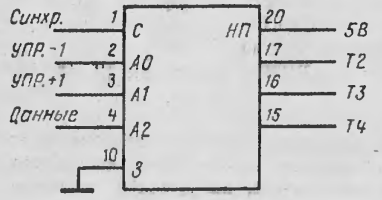


Рис. 2. Схема реверсивного сдвигающего регистра на ПМЛ

Таблица 3

Таблица истинности БИС КР1556ХП6

КР1556ХП6		А0	А1	А2	А3	А4	А5	А6	А7	В0	Т1	Т2	Т3	Т4	Т5	Т6	В7
	0	X	X	X	1	1	1	X	X	X	X	X	X	X	X	X	X
	1	1	0	1	0	1	1	1	0	X	X	X	X	X	X	X	X
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В0	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	9	0	1	0	1	1	X	X	X	X	X	X	X	X	X	X	X
	10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Т1	12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
. . . . .																	
	57	X	X	1	1	X	X	X	X	X	X	X	X	X	X	X	X
	58	1	0	1	0	0	0	0	0	1	0	1	0	X	X	X	X
	59	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	60	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В7	61	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Уравнения имеют следующий вид:

С А0 А1 А2 А3 А4 А5 А6 А7 З
ВК / В7 / Т6 / Т5 / Т4 / Т3 / Т2 / Т1 / В0 НП
/ В0 = А0×/А1×А2×/А3×А4×
×А5×А6/А7
/В0 = (А3×А4×А5)
/Т1 = /А0×А1×/А2×А3×А4
/В7 = А0×/А1×А2×/А3×/А4×/
А5×/А6×/А7×В0×/Т1×Т2×/Т3
/В7 = (А2×А3)

Пример проектирования

Представление данных в виде систем булевых уравнений позволяет разработчику проектировать различные цифровые устройства на качественно более высоком уровне.

Реверсивный сдвигающий регистр на D-триггерах (рис. 1).

Условные обозначения, использованные на схеме, соответствуют следующим обозначениям КР1556ХП4:
УПР. $-1 \equiv$ А0; УПР. $+1 \equiv$ А1; Данные $\equiv$ А2; Синхр. $\equiv$ С.
Запишем систему уравнений:

/Т2 = А0 × / Т3+А2 × А1
/Т3 = А0 × /Т4+ / Т2 × А1
/Т4 = А0 + /Т3 × А1

После программирования микросхемы получаем прибор, схема которого представлена на рис. 2.

Сравнение структур регистра и схемы КР1556ХП4 обнаруживает, что при реализации регистра большая часть ресурса схемы может быть использована для построения других цепей. В ряде применений БИС ПМЛ позволяют сократить число микросхем средней степени интеграции в несколько раз.

Телефон 468-81-75, Москва

Статья поступила 26.01.88

УДК 681.3.06

В. М. Чурков, С. А. Котрелев, Л. Е. Захарова

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ САПР ЦИФРОВЫХ УСТРОЙСТВ ДЛЯ ПРОГРАММИРУЕМОЙ МАТРИЧНОЙ ЛОГИКИ СЕРИИ КР1556

Структура программного обеспечения САПР ПМЛ

Программное обеспечение (ПО) функционирует в ОС ДВК. Для реализации трех функций САПР — проектирования, программирования и тестирования — использован модульный принцип построения программы. Все три функции САПР для ПМЛ выполнены в виде набора подпрограмм, обращение к которым происходит из основной программы. Функция проектирования микросхемы представлена шестью режимами: набора данных, набора данных в виде уравнений, записи файла, чтения файла, распечатки таблицы и ввода данных с эталона. Функция программирования — тремя режимами: теста аппаратуры, входного контроля перемычек, программирования. Функция тестирования — одним режимом сравнения с таблицей данных. Все режимы реализованы в виде отдельных подпрограмм. В основной программе предусмотрены два режима: монитора и возврата в систему.

Модульный принцип построения программы имеет ряд преимуществ: каждую подпрограмму с минимальной дора-

боткой можно использовать в ПО других САПР, кроме того, при необходимости можно ввести любой дополнительный режим.

Поскольку в САПР одновременно реализуется только один режим, то существует возможность построения ПО по оверлейному принципу, т. е. в памяти ДВК постоянно находятся основная программа, часть подпрограмм и подчиненные подпрограммы, используемые в нескольких подпрограммах, а все остальные подпрограммы располагаются на диске.

При необходимости нужная подпрограмма загружается в память и с ней можно работать.

В ПО САПР для ПМЛ большой объем занимает подпрограмма набора данных в виде уравнений — первый оверлейный сегмент оверлейной области. Второй оверлейный сегмент включает подпрограммы теста аппаратуры, ввода данных с эталона, входного контроля перемычек, сравнения с таблицей данных и программирования. Все остальные подпрограммы и основная программа находятся в корневом сегменте и постоянно присутствуют в памяти ДВК. При необходимости увеличить число режимов оверлейная структура может быть усложнена.

Модульный принцип построения программы позволяет подготовить подпрограммы на различных языках программирования, в данном случае — на Фортране и макроассемблере. Подпрограммы, реализующие различные режимы программатора, могут также иметь подчиненные подпрограммы (на усмотрение программиста). У подпрограмм, написанных на Фортране, подчиненные подпрограммы могут быть на Фортране и на макроассемблере, у подпрограмм на ассемблере — только на ассемблере. Данные в подпрограммы, написанные на Фортране, передаются через общие области COMMON для экономии памяти ДВК. Данные в подпрограммы, написанные на ассемблере, передаются стандартным образом через регистр R5*. Через него можно передать также фактические параметры и начальный адрес таблицы программирования.

Пример 1. Обращение к подпрограмме, определяющей код вводимого с клавиатуры дисплея символа. Программа на Фортране анализирует код символа, получив его через подпрограмму на ассемблере. Ниже приведен фрагмент программы на Фортране

```
  2 CALL ID (N)
    IF (N. NE.127) GO TO 1
```

и подпрограмма на ассемблере

```
ID:: TSTB     177560
     BPL      ID
     MOV      177562, @2 (R5)
     RTS      PC
     .END     ID
```

Командой MOV 177562,@2 (R5) в программу передается фактическое значение параметра N.

Пример 2. Обращение к подпрограмме программирования. В основной программе описан массив JD таблицы программирования. Начальный адрес таблицы передается в подпрограмму на ассемблере.

Фрагменты программы на Фортране

```
DIMENSION JD(512) . . . . . .
. . . . . .
IF (M.EQ.304) CALL PROG (JD(1))
```

Начало подпрограммы на ассемблере

```
        .GLOBL    TAB, IPE, PR, PR2
PROG::  ADD       #2, R5
        MOV       (R5), R0
```

Таким образом, в регистре R0 находится адрес начала таблицы программирования. В подпрограмму на ассемблере из программы на Фортране можно передать любое число параметров. В регистре R5 передается адрес списка аргументов, формат которого изображен на рис. 1.

Подпрограммы на ассемблере, используемые программой на Фортране, могут иметь собственные подчиненные подпрограммы, но не макроопределения. Подчиненные подпрограммы должны быть описаны в подпрограмме оператором .GLOBL (как в подпрограмме программирования). Макроопределения, в том числе стандартные, в подпрограммах на ассемблере не используются. Например, для вывода текста на дисплей в подпрограмме на ассемблере написана подпрограмма IPE вместо стандартного макроопределения .PRINT.

Починенная подпрограмма вывода текста на дисплей

```
IPE ::    CLR     R2
          MOV     (R0), R2
          ADD     #2, R0
IR:       TSTB    177564
          BPL     IR
          MOV     (R0), 177566
          ADD     #2, R0
          DEC     R2
          BNE     IR
          RETU    RN
          .END    IPE
```

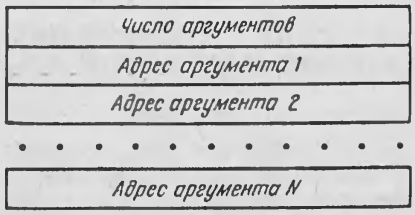


Рис. 1. Формат списка аргументов при передаче данных из программы на Фортране в подпрограмму на ассемблере

К моменту обращения к подпрограмме вывода на дисплей регистры R0 и R2 должны быть свободны.

Фрагмент подпрограммы на ассемблере с обращением к подчиненной подпрограмме IPE

```
          .GLOBL    TAB, IPE, PR,
                    PR2
PROG :: ADD         #2, R5
. . . . . .
          MOV       #PPK, R0
          CALL      IPE
67¤:      RTS       PC
PPK:      .WORD     55, 12, 12, 15, 16,
                    40, ...
```

В директиве .WORD первым идет восьмеричное значение числа символов, выводимых на дисплей, а потом восьмеричные значения самих символов.

Возвращение в программу на Фортране из подпрограмм на ассемблере происходит только по команде RTS PC; обращение к подчиненным подпрограммам из подпрограмм на ассемблере — по команде CALL IPE, а возвращение — по команде RTS PC.

Для отладки режимов программирования, входного контроля, теста аппаратуры, ввода данных с эталона и сравнения с таблицей данных была использована специальная подчиненная подпрограмма:

```
A :: TSTB     177560
     BPL      A
     CMP      #157, 177562
     BNE      A
     RTS      R4
     .END     A
```

В нужном месте подпрограммы на ассемблере, где необходимо остановить программу и проверить напряжения на ячейках программатора, надо вставить оператор JSR R4, A. Программа будет ждать подачи с клавиатуры русской буквы «О» (восьмеричный код 157). После проверки напряжений надо нажать клавишу «О», и программа будет выполняться дальше до следующей команды JSR R4, A. Регистр R4 к моменту обращения к подчиненной подпрограмме A должен быть свободен. После отладки все команды JSR R4, A и подпрограмму A надо удалить из программы.

Во всей программе нет обращения к конкретным ячейкам, кроме ячеек, находящихся в программаторе. Если регистров не хватает, то используются ячейки с меткой на директиве .WORD. Например:

```
          MOV # BL, R0
          MOV # 5, R3
1¤:       MOV# 1, (R0) +
          DEC  R3
          BNE  1¤
. . . . . .
BL :      .WORD,,,,,
```

Во все пять ячеек с меткой BL за-

*Программное обеспечение ДВК. Операционная система с разделением функций ОС ДВК. Фортран. Руководство системного программиста.— Т. 5. Кн. 2. Ч. 3.

сылаются единицы. При наличии оверлейной структуры программы конкретными ячейками лучше не пользоваться.

Применение уравнений в САПР ПМЛ для проектирования микросхем

Из-за больших размеров таблицы истинности ПМЛ (16×64=1024) и сложности ее составления целесообразно вместо таблицы использовать уравнения. Приведенные методики позволяют анализировать, редактировать и переводить уравнения в таблицу истинности и обратно.

Данные в таблице могут находиться в следующих состояниях:

1) в конъюнкцию включается прямое значение переменной (при этом в соответствующий разряд необходимо записать 1);

2) в конъюнкцию включается инверсное значение переменной (записывается 0);

3) переменная в конъюнкцию не включается (набирается символ X);

4) если переменная не запрограммирована, то на экране отображается символ «—» и конъюнкция не включается в выходную функцию.

Переменные обозначаются тремя буквами: A, B, T. Переменные A0...A9 — входы, B0...B7 — выходы-входы, T0... T7 — выходы.

Ниже представлены примеры уравнений, соответствующих таблицам истинности (табл. 1...4).

Для схемы КР1556ХЛ8:

/B0=A0×/A1×A3+A4×/A1×/A7 ×
×B5×/A9+/A0×/A1×/A2 Тип 1

/B0=(A0×A5×/A6×/A7×A1×/B3)
Тип 2

/B3=/A7×/A6+A0+/A3+A5×A7

Для схемы КР1556ХП8:

/T0:=A0×A1+A2×/A3+A4×A5
+A6×A7+A0+A1+A2+A3×A2
/T1:=A0×/T6×T7×A1

Для схемы КР1556ХП6:

/B0=A0×T2×B7×T4
/T1:=A6×A4

Для схемы КР1556ХП6:

/T2:=A1×A7+/B7
/B1=(A4×A5)
/B0=(A1×A7)
/BO=/B7

Уравнения типа 1 описывают B- или T-выходы. Каждое слагаемое представляет одну конъюнкцию. Для B-выхода может быть семь слагаемых, а для T-выхода — восемь. Символ «/» перед переменной означает, что переменная входит в конъюнкцию с инверсным значением, переменная без символа «/» — с прямым значением; символ «×» объединяет переменные, описывающие одну конъюнкцию. Сомножители, описывающие одну конъюнкцию, не повторяются.

Уравнение типа 2 описывает конъ-

Таблица 1

Таблица истинности БИС КР1556ХЛ8

КР15	56ХЛ8	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	B1	B2	B3	B4	B5	B6
	1	1	1	X	X	X	1	0	0	X	X	X	X	0	X	X	X
	2	1	0	X	1	X	X	X	X	X	X	X	X	X	X	X	X
	3	X	0	X	X	1	X	X	0	X	0	X	X	X	X	1	X
	4	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X
B0	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
B1	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	. .	. .		.	.		.		.	.	.	.					
	57	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	58	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	59	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	60	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
B7	61	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	64	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Таблица 2

Таблица истинности БИС КР1556ХП8

КР1556ХП8	A0	A1	A2	A3	A4	A5	A6	A7	T0	T1	T2	T3	T4	T5	T6	T7
1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	X	1	0	X	X	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	1	1	X	X	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	1	1	X	X	X	X	X	X	X	X
T0 5	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
6	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X
8	X	X	1	1	X	X	X	X	X	X	X	X	X	X	X	X
9	1	1	X	X	X	X	X	X	X	X	X	X	X	X	0	1
10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
T1 13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
. . . . .																
57	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
58	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
59	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
T7 60	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
61	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
64	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Таблица 3

Таблица истинности БИС КР1556ХП6

КР1556ХП6		А0	А1	А2	А3	А4	А5	А6	А7	В0	Т1	Т2	Т3	Т4	Т5	Т6	В7
	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
	2	1	X	X	X	X	X	X	X	X	X	1	X	1	X	X	1
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В0	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	9	X	X	X	X	1	X	1	X	X	X	X	X	X	X	X	X
	10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Т1	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
. .	. .			.	.		.	.	.		.	.		.	.		
	57	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	58	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	59	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	60	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В7	61	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	64	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Таблица 4

Таблица истинности БИС КР1556ХП4

КР1556ХП4		А0	А1	А2	А3	А4	А5	А6	А7	В0	В1	Т2	Т3	Т4	Т5	Т6	Т7
	1	X	1	X	X	X	X	X	1	X	X	X	X	X	X	X	X
	2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В0	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	9	X	X	X	X	1	1	X	X	X	X	X	X	X	X	X	X
	10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В1	13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
. .	. .																
	57	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	58	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	59	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	60	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
В7	61	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	62	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	63	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	64	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

юнкцию, управляющую выходным состоянием В-выхода. В скобках могут находиться сомножители, описывающие только одну конъюнкцию.

Левая часть уравнений для В-выходов отделяется от правой символом «=», а для Т-выходов — символами «:=».

При переводе уравнений в таблицу истинности переменные с инверсным значением переводятся в 0, с прямым — в 1. Отсутствующая в произведении переменная переводится в символ X. Если в уравнении для Т-выхода число слагаемых меньше восьми, то не описанные в нем конъюнкции не пережигаются. Если в уравнении для В-выхода число слагаемых меньше семи, то не описанные конъюнкции также не пережигаются, за исключением конъюнкции, управляющей выходным состоянием. Если для данного В-выхода есть уравнение типа 1, но нет уравнения типа 2, то все перемычки в конъюнкции, управляющей выходным состоянием, пережигаются. Если в уравнениях В- или Т-выходы не описаны, то все конъюнкции этого выхода не пережигаются.

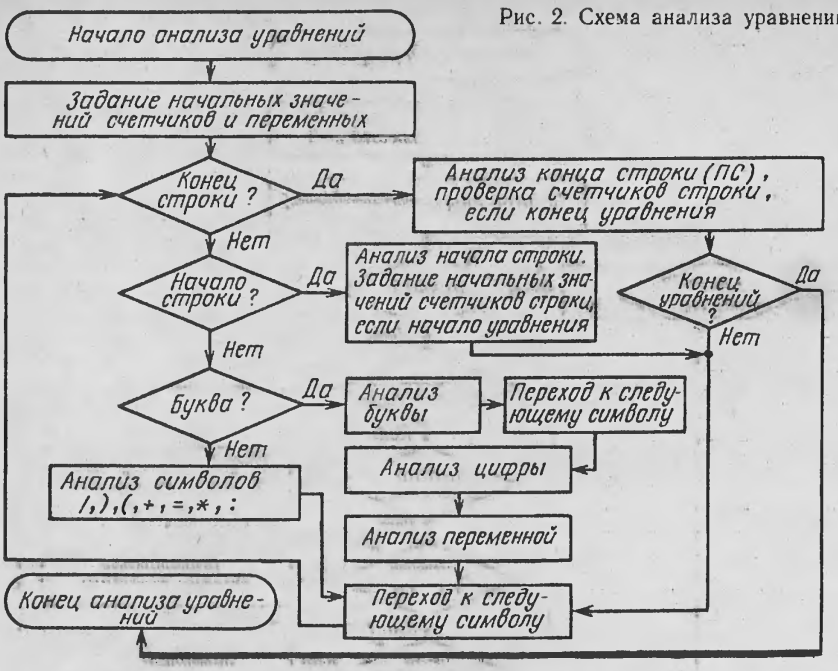
Программа классифицирует ошибки по следующим типам:

1. Неверный символ.
2. Неверный знак операции, лишняя или недостающая скобка.
3. В произведении повторяются операторы.
4. Знаков «+» больше шести при В-выходе или знаков «+» больше семи при Т-выходе.
5. В скобках встретился знак «+».
6. Дважды описан один и тот же выход. Если второй раз В-выход описан через скобки,— это не ошибка.
7. Т-выход описан через скобки.

При анализе уравнений (рис. 2) происходит последовательный перебор всех символов в массиве уравнений L (4580). При анализе символа рассматриваются соседние с ним символы (табл. 5).

При проходе уравнений ведутся счетчики номеров символа в массиве уравнений и в строке, а также счетчики уравнений для определения числа ошибок, проверки правильности набранных переменных, обнаружения уравнений, описывающих один и тот же выход, и т. д. В начале каждого уравнения формируются счетчики строки для обнаружения лишней или недостающей скобки и повторяющихся переменных в произведении, подсчета числа знаков «+» в уравнении и т. д. В начале каждого уравнения всем счетчикам строки придаются начальные значения; в конце уравнения счетчики проверяются и на дисплей выводятся данные об ошибках.

Строки отделяются символом (ПС). Перед заполнением кодами символов массив L обнуляется. После заполнения за пределами конца строки в массиве остаются одни нули. Одно уравнение может занимать несколько строк, но в одной строке — только одно уравнение. Максимальный размер мас-



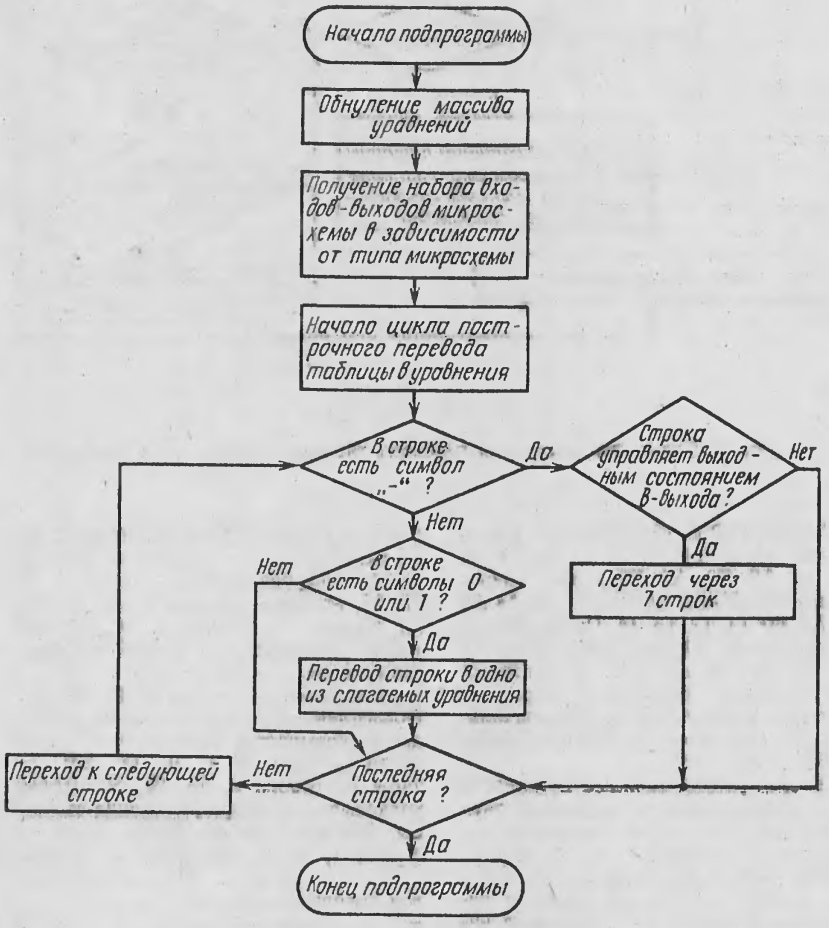


Рис. 3. Схема подпрограммы перевода уравнений в таблицу

Таблица 5

Таблица проверки символов

Символ	Соседние символы (местонахождение символа)
=	В четвертой или пятой позиции с начала уравнения, после цифры или «:» перед буквой или «(»
:	В четвертой позиции с начала уравнения после цифры перед «=»
/	После «(», «=», «*», «+» перед буквой, если не в первой позиции. Если в первой позиции, то в начале уравнения или после «)» и цифры перед буквой
*, +	После цифры перед буквой и «/»
(	Только в уравнениях для В-вывода, в пятой позиции после «=» перед буквой и «/»
)	Только в уравнениях для В-выхода, после цифры перед «/» или нулем (концом уравнений)
Буква A, B, T	После «/», «(», «=», «*», «+» перед цифрой
Цифра	После буквы перед «)», «*», «+», «/» и нулем (концом уравнений)

сива уравнений L определяется из расчета, что в них описаны все выходы схемы и в каждом уравнении присутствуют все возможные переменные с инверсным значением.

При наборе уравнений с клавиатуры набираются следующие символы: A, B, T, 0...9, :, =, *, +,), (,/,ПС. Следовательно, при переводе уравнений из таблицы истинности других символов, кроме вышеперечисленных, в массиве уравнений не будет. При анализе символов происходит подсчет числа ошибок, классификация ошибок по типам и вывод на дисплей данных об ошибках. Уравнения переводятся в таблицу после анализа в отсутствие ошибок (рис. 3). Перевод состоит в определении номеров строки и столбца каждой переменной уравнения и в заполнении разряда таблицы соответствующим символом 0 или 1. В начале перевода вся таблица заполняется символами «—». Если определен номер строки для какой-то переменной и такого номера раньше не было, то строка заполняется символами X. Символы 0 или 1 появляются уже потом в необходимых разрядах строки (в остальных разрядах остаются символы X). В начале любого уравнения по переменной слева от символа «=» определяется номер строки, заполняемой первой. Например, для переменных TI (I=0, ..., 7) номер первой строки равен (I—1)×8+ +1. При обнаружении в уравнении знака «+» номер строки увеличивается

на 1 и так до начала следующего уравнения. Для переменной B1 (I=0, ..., 7) номер первой строки определяется аналогично. Однако если B-выход описан уравнением типа 1, то заполнение начинается со второй строки. Причем, если к этому моменту первая строка не заполнена (не было уравнения типа 2 для этого B-выхода), то она заполняется символами X. Если в дальнейшем уравнение типа 2 для этого B-выхода встретится, то нужные разряды первой строки этого B-выхода заполнятся 0 или 1, а в остальных разрядах останется символ X.

По переменным справа от символа «=» определяются номера столбцов. Например, для переменных AI (I==1, ..., 9) номер столбца равен (I+1). Для микросхемы КР1556ХЛ8 в переменных BI (I=1, ..., 6) номер столбца равен (I+10). Для остальных микросхем (КР1556ХП8, КР1556ХП6, КР1556ХП4) в переменных TI (I=0, ..., 7) и BI (I=0, 1, 6, 7) номер столбца равен (I+9). Разряд в таблице для переменных с символом «/» заполняется нулями, без символа «/» — единицами.

Перевод таблицы истинности в уравнения осуществляется только в режиме ввода данных с эталона (рис. 4). Это сделано для пользователей, желающих работать только с уравнениями, без таблиц. Необходимо помнить, что при переводе таблиц в уравнения структура таблиц теряется, сохраняется только логика работы микросхемы. Например, если в строке (конъюнкции) таблицы истинности имеется хотя бы один символ «—», то при переводе в уравнения считается, что вся строка состоит из символов «—». Если вся строка состоит из символов X, то при переводе в уравнения считается, что она состоит из символов «—». Если в строке, управляющей выходным состоянием B-выхода, есть хотя бы один символ «—», то все восемь строк, описывающих B-выход, считаются заполненными символами «—». В режиме программирования и других, связанных непосредственно с программатором, таблица истинности переводится в «машинную» таблицу, т. е. в массив JD(512). Подпрограмма IG переводит машинную таблицу в таблицу истинности, а подпрограмма GI — таблицу истинности в машинную таблицу. Размерность машинной таблицы (массив JD) — 512 чисел. Каждое число массива содержит в младших четырех разрядах информацию о четырех перемычках: единица в разряде числа соответствует целой перемычке, а нуль — пережженной. Первоначально в чистой микросхеме во все члены массива JD записано восьмеричное число 17.

При переводе таблицы истинности в машинную столбцы таблицы истинности меняются местами. Порядок столбцов для микросхемы КП1556ХЛ8 «A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 B1 B2 B3 B4 B5 B6» изменяется: «A0 A8 A1 B1 A2 B2 A3 B3 A4 B4 A5 B5 A6 B6 A7 A9», для микросхемы КР1556ХП8 «A0 A1 A2 A3 A4 A5 A6 A7 T0 T1 T2 T3 T4 T5 T6 T7» — на «A0 T0 A1 T1 A2 T2 A3 T3 A4 T4 A5 T5 A6 T6 A7 T7», для микросхемы КР1556ХП6 «A0 A1 A2 A3 A4 A5 A6 A7 B0 T1 T2 T3 T4 T5 T6 B7» — на «A0 B0 A1 T1 A2 T2 A3 T3 A4 T4 A5 T5 A6 T6 A7 B7». Измененный вид таблицы истинности отражает структуру микросхемы, но первоначальный вид таблицы истинности более удобен для работы.

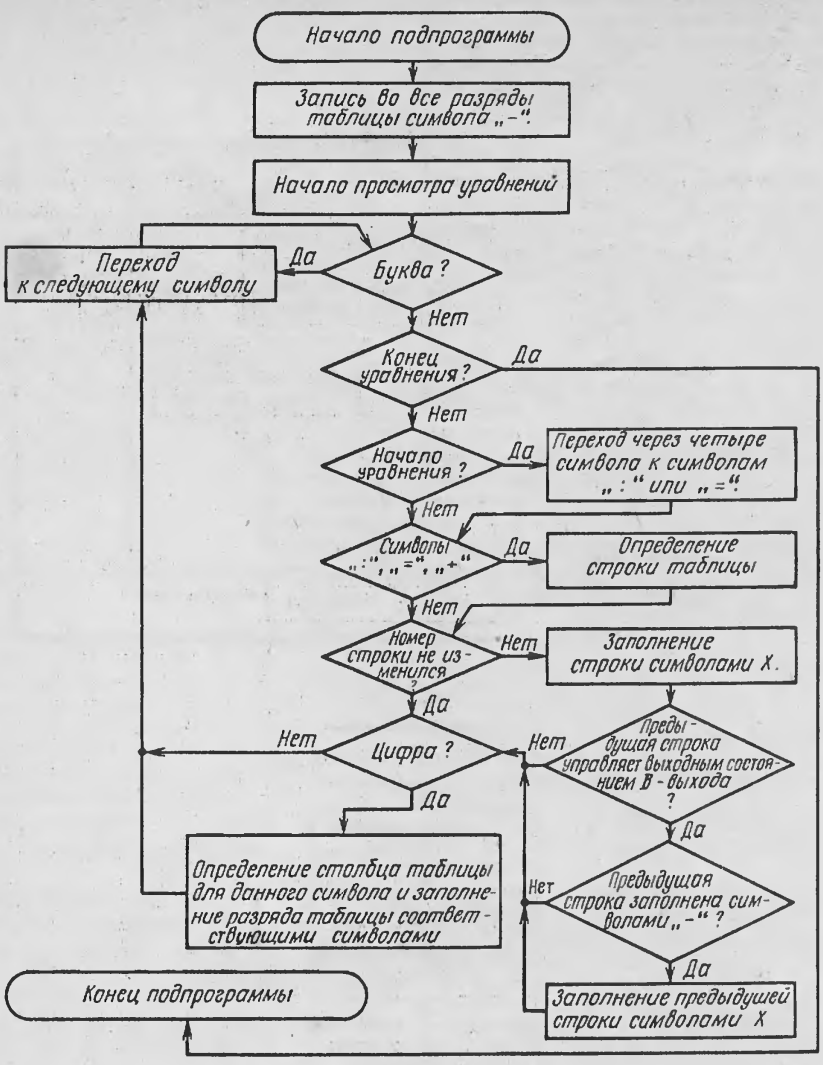


Рис. 4. Схема подпрограммы перевода таблицы истинности в уравнения

Каждый разряд таблицы истинности несет информацию о состоянии двух перемычек. Обе целые перемычки соответствуют символу «—», обе пережженные — X, пережженная левая и целая правая — 0, целая левая и пережженная правая — 1. Упаковка измененной таблицы истинности в машинную таблицу происходит по столбцам: сначала строки с 1 по 32, потом с 33 по 64. Первым упаковывается первый столбец: первое число — строки 1, 9, 17, 25; второе — строки 2, 10, 18, 26; ...; восьмое — строки 8, 16, 24, 32. Далее упаковывается второй столбец: девятое число — строки 1, 9, 17, 25; ...; шестнадцатое — строки 8, 16, 24, 32. Последним из первой половины таблицы упаковывается 32-й столбец: 249-е число — строки 1, 9, 17, 25; ...; 256-е — строки 8, 16, 24, 32. Далее упаковывается вторая половина таблицы. Первый столбец: 257-е число — строки 1, 9, 17, 25; 258-е — строки 2, 10, 18, 26 и т. д., как для первой половины таблицы.

Телефон 468-81-75, Москва

Статья поступила 26.01.88

УДК 681.326.3

В. М. Чурков, Л. Е. Захарова

ПОДСИСТЕМА ПРОЕКТИРОВАНИЯ ПЛМ

Подсистема проектирования цифровых устройств на серийно выпускаемых ПЛМ К556РТ1, К556РТ2 функционирует на ДВК2М в операционной системе ОС ДВК. Программное обеспечение подсистемы обеспечивает работу в 14 режимах: МО — монитор; ПР — программирование; СЭ — сравнение с эталоном; ЛК — логический контроль; ТЕ — тест аппаратуры; ЗФ — запись файла; ОТ — отладка; ЗЭ — запись с эталона; ДА — набор данных; ФК — фукциональный контроль; КО — входной контроль; РТ — распечатка таблицы; ЧФ — чтение файла; ВС — возврат в систему.

Эти режимы осуществляют четыре функции управления: проектирование, проверку, программирование и тестирование ПЛИС.

Для проверки исправности программатора служит режим «тест аппаратуры». В режиме «монитор» на экране печатается список всех режимов. Режим «возврат в систему» используется по окончании работы программатора для возврата в систему ОС ДВК. Процесс проектирования состоит из пяти режимов: «набор данных», «распечатка таблицы», «запись файла», «чтение файла» и «запись с эталона». Режим «набор данных» является входным языком программирования и позволяет редактировать и заполнять заново таблицу истинности ПЛМ (табл. 1). При выполнении режима таблица истинности ПЛМ по страницам выдается на дисплей. С помощью клавиши «ПС» таблица листается вперед, клавиши «ГТ» — назад. Клавиши «⇑», «⇓», «→», «←» позволяют подвести риску к любой позиции таблицы.

Таблица 1

Таблица истинности ПЛМ

Терм	Входная переменная															Выходы ПЛМ							
	15	14	13	12	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	—	А	—	—	—	—	А	А
2	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Х	Х	—	А	—	А	А	А	—	—
3	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	В	—	—	—	—	—	—	—	А
4	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Х	Н	Н	Х	—	А	—	А	А	А	—	А
5	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Н	Н	А	—	А	—	—	А	А	А
6	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	В	Н	А	А	А	А	А	—	—	—
7	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	В	А	—	—	—	—	—	—	—
8	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Н	—	—	А	А	—	—	А	—
9	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Х	В	В	—	—	А	—	—	—	А	А
10	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	Н	—	А	—	А	—	А	А	А
11	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Н	В	А	А	А	А	А	—	—	—
12	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	В	Н	—	—	—	А	А	—	—	—
13	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	Н	Н	Х	—	А	А	А	А	А	А	А
14	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Н	Н	Н	А	—	А	А	А	А	А	А
15	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	Н	В	В	А	А	—	А	А	А	А	А
16	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	Н	В	Н	А	А	А	—	А	А	А	А
17	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Н	В	Н	А	А	А	А	—	А	А	А
18	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	В	В	В	А	А	А	А	А	—	А	А
19	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	В	Н	Н	А	А	А	А	А	А	—	А
20	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	В	Н	В	А	А	А	А	А	А	А	—
21	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	В	В	Н	—	—	—	—	—	—	А	—
22	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	Н	В	В	В	—	—	—	—	—	А	—	—
23	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Н	Н	В	—	—	—	—	А	—	—	—
24	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	В	В	Х	Х	Н	—	—	—	А	—	—	—	—
25	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	А	А	А	А	А	А	А	А
26	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Таблица истинности состоит из выходов ПЛМ, входных переменных (конъюнкторов), выходной функции (дизъюнкторов). Выходы ПЛМ редактируются только на первой странице таблицы и имеют два значения — Н и В, входные переменные — значения Н, В, Х, и «—», выходная функция — значения «—» и А. Программное обеспечение построено так, что других значений набрать нельзя. Таблица истинности удобна для набора и редактирования данных. Для работы с программатором есть еще две таблицы — контроля и данных (программирования). Эти две таблицы имеют вид, удобный для непосредственной работы с микросхемой. Данные для программирования находятся в таблице данных. В начале режима «набор данных» таблица данных переводится в таблицу истинности; в конце режима измененная таблица истинности переводится в таблицу данных (табл. 2). В режиме «запись файла» файл данных переводится на диск, создается библиотека файлов. Таблицу истинности после набора можно записать в режиме «запись файла», а при необходимости прочитать в режиме «чтение файла». Режим «запись с эталона» позволяет записать информацию с эталонной микросхемы в таблицу данных, посмотреть в таблице истинности, записать на диск в режиме «запись файла» или использовать для программирования.

В режиме «отладка» проверяется правильность составления таблицы истинности для решения какой-либо задачи до программирования микросхемы на математической модели ее работы. По входному вектору, заданному с терминала, определяется выходной вектор и термы, которые участвуют в формировании выходного вектора, задаваемого в восьмеричном коде. Входной вектор задается в двоичном коде, причем, кроме 0 и 1, любой бит входного вектора может иметь безразличное значение Х. Для всех входных векторов, которые можно получить с учетом битов с безразличными состояниями, определяются выходные векторы и формирующие их термы (табл. 3). В процессе отладки таблицы истинности задается входной вектор в виде ХХХХХХХХХХХХХХХХ и по нему формируются данные обо всех входных векторах от 0 до $2^{16}-1$. Выдача большо-

Таблица 2

Таблица данных ПЛМ

0	177 777	3	177 777	7	177 770	11	177 777	14	177 763
12	177 765	5	177 772	6	177 771	17	177 764	4	177 773
15	177 762	16	177 761	21	177 757	30	177 747	23	177 754
22	177 755	32	177 745	37	177 740	24	177 753	25	177 752
26	177 751	27	177 750	31	177 746	36	177 747	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	103	134	1	135
247	370	200	62	43	127	370	30	177	277
337	357	367	373	375	376	2	4	10	20
377	377	377	377	377	377	377	377	377	377
377	377	377	377	377	377	377	377	377	377
377	377	377	377	377					

Таблица 3

Отладка проекта микросхемы

Входной вектор 0000000000XXXXX	Выходной вектор	Термы	
0	137	1	2 4
1	135	2	4
2	134	2	
3	134	2	
4	127	10	
5	200	7	
6	62	8	
7	1	3	
10	135	4	
11	135	4	
12	370	6	
13	43	9	
14	247	5	
15	370	11	
16	30	12	
17	43	9	
20	177	13	
21	177	13	
22	357	16	
23	337	15	
24	375	19	
25	376	20	
26	2	21	
27	4	22	
30	277	14	24
31	10	23	
32	367	17	24
33	0		
34	20	24	
35	0		
36	20	24	
37	373	18	

го объема информации на экран в режиме «отладка» происходит по страницам. Для просмотра следующей страницы надо нажать клавишу «ВК», для досрочного выхода из режима в конце страницы — букву «К» и клавишу «ВК». При окончании выдачи информации на экран режим заканчивается.

Программирование осуществляется в режимах: «входной контроль» и «программирование». В режиме входного контроля данные записываются с микросхемы в таблицу контроля и анализируются. Если микросхема была чистой, т. е. все перемычки целые, то на дисплей выдается сообщение «годен» и режим заканчивается. В противном случае, таблица контроля переводится в таблицу истинности, которая постранично выдается на экран. Чтобы посмотреть, что же конкретно записано в микросхему и при этом не испортить таблицу данных, можно воспользоваться таблицей контроля в режиме входного контроля.

Режим программирования осуществляется в соответствии с таблицей данных. На всей продолжительности режима (до 50 с) на экран выдается сообщение «режим выполняется», при отрицательном результате программирования — сообщение «брак», при положительном — «годен».

Режим организован так, что микросхему можно допрограммировать. В начале режима таблица данных анализируется на корректность, т. е. проверяется, нет ли необходимости «восстановить» какую-то пережженную перемычку. Если таблица истинности составлена некорректно, без учета конкретного состояния перемычек, то на экран выдается сообщение «некорректность данных», режим заканчивается.

Тестирование микросхемы осуществляется в режимах: «сравнение с эталоном», «логический контроль», «функциональный контроль». Даже при положительном результате программирования в режиме сравнения с эталоном может быть получен отрицательный результат из-за разбрызгивания металла во время программирования, или внутреннего дефекта пережигаемой микросхемы. При положительном результате сравнения на экран выдается сообщение «годен», при отрицательном — «брак». Режим «логический контроль» позволяет пользователю проверить работу запрограммированной микросхемы.

Режим «функциональный контроль» позволяет проверить все 2^{16} адресов микросхемы, сравнивая их с данными математической модели микросхемы. Максимальная продолжительность режима 7 мин (в среднем 4 мин). При положительном результате сравнения данных ПЛМ с данными ЭВМ в конце режима на дисплей выдается текст «годен». При отрицательном результате сравнения на экран выдается адрес несравнения, данные ЭВМ и ПЛМ, по этому адресу. Если адресов несравнения много, то информация о них выдается на экран постранично.

Телефон 468-81-75, Москва

Статья поступила 26.01.88

УДК 681.326+681.325.5

В. М. Чурков, С. А. Котрелев

ПОДСИСТЕМА ПРОЕКТИРОВАНИЯ НА ЭПЛК

Подсистема предназначена для проектирования и изготовления устройств на базе электрически программируемого логического контроллера (ЭПЛК), представляющего собой программируемую пользователем БИС микросхему КР556РТ3.

Рассмотрим некоторые функциональные особенности микросхемы (рис. 1) КР556РТ3. Микросхема состоит из двух программируемых матриц: И и ИЛИ. Матрица И включает в себя 48 схем И, каждая из которых имеет 45 входов: 32 входа от 16 входных переменных, 12 входов с регистра обратной связи и вход матрицы обращения. Матрица ИЛИ содержит 29 схем ИЛИ, каждая из которых связана со всеми выходами матрицы И. Таким образом, каждая схема ИЛИ имеет 48 входов. 28 выходов схем ИЛИ соединены со входами установки и сброса 14 RS-триггеров, образующих 8-разрядный выходной регистр и 6-разрядный регистр обратной связи; выход 29 схемы ИЛИ — со входом матрицы обращения. Установка или сброс RS-триггеров синхронизируется внешним сигналом С. В микросхеме имеется также перемычка, определяющая функцию входа ВК/PR: при целой перемычке этот вход выполняет функцию предустановки триггеров, при оборванной перемычке — функцию управления третьим состоянием выходов F0...F7. Таким образом, микросхема КР556РТ3 реализует 8 функций от 16 входных переменных. Наличие выходного регистра и регистра обратной связи определяет функциональное назначение ЭПЛК как контроллера последовательностей.

Из описания структуры ЭПЛК видно, что эта микросхема является сложным устройством и для ее применения на практике необходимы специальные средства. Одно из таких средств —

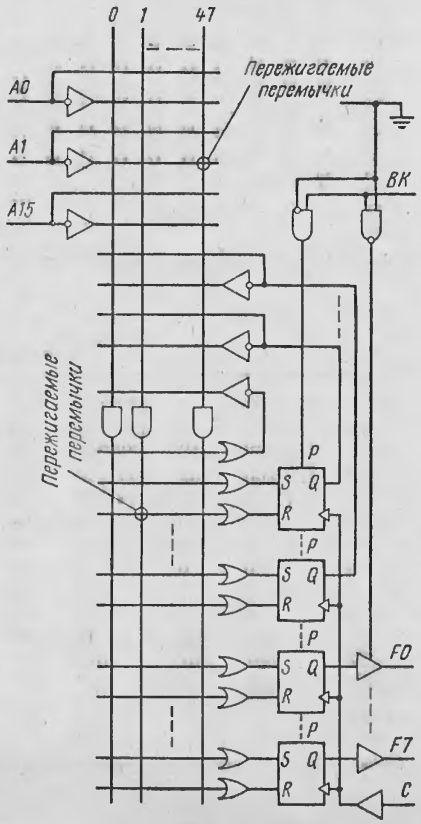


Рис. 1. Структурная схема БИС КР556РТ3

ПРОГРАММАТОР БИС ЭПЛК

МОН	МОНИТОР
ВК	ВХОДНОЙ КОНТРОЛЬ
ЗЭ	ЗАПИСЬ С ЭТАЛОНА
ДАН	НАБОР ДАННЫХ
ПРОГ	ПРОГРАММИРОВАНИЕ
СЭ	СРАВНЕНИЕ С ЭТАЛОНОМ
КП	КОНТРОЛЬ ПОСЛЕДОВАТЕЛЬНОСТЕЙ
ВС	ВОЗВРАТ В СИСТЕМУ
ФК	ФУНКЦИОНАЛЬНЫЙ КОНТРОЛЬ
ЛК	ЛОГИЧЕСКИЙ КОНТРОЛЬ
РТ	РАСПЕЧАТКА ТАБЛИЦЫ
ТЕ	ТЕСТ АППАРАТУРЫ
ЗФ	ЗАПИСЬ ФАЙЛА
ЧФ	ЧТЕНИЕ ФАЙЛА
ПГ	ПРОГНОЗИРОВАНИЕ

Рис. 2. Программатор БИС ЭПЛК

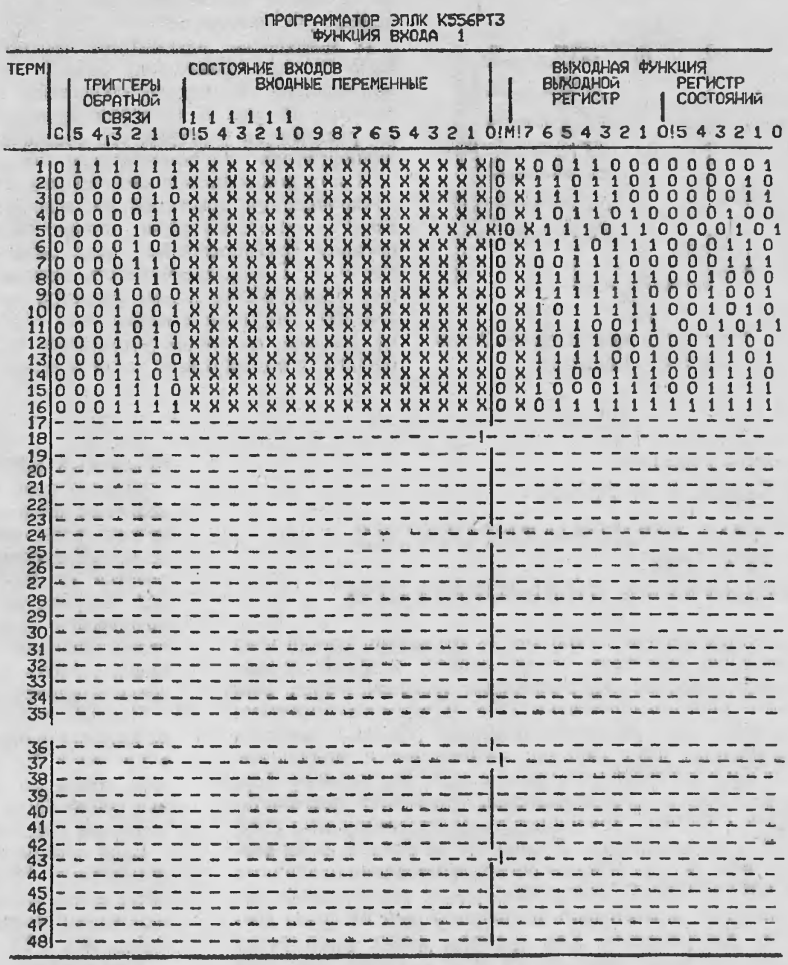


Рис. 3. Таблица данных для БИС ЭПЛК

подсистема проектирования на ЭПЛК. В состав подсистемы входят вычислительный комплекс ДВК2М, работающий под его управлением программатор ЭПЛК и пакет программ, записанных в виде отдельного файла на ГМД. Подсистема обеспечивает выполнение ряда режимов работы (рис. 2).

В режиме ТЕСТ АППАРАТУРЫ автоматически проверяется работоспособность программатора. В случае его исправности на экране монитора появляется сообщение «Готов к работе», а в случае неисправности — номера отказавших узлов или блоков программатора.

В режиме ВХОДНОЙ КОНТРОЛЬ предварительно проверяется микросхема. Если все пережигаемые перемычки целы, на экране распечатывается сообщение «Микросхема незапрограммирована», иначе, распечатывается таблица с координатами пережженных перемычек.

В режиме ЗАПИСЬ С ЭТАЛОНА данные программирования с эталонной микросхемы переписываются в ОЗУ ЭВМ.

Данные программирования можно также набрать с помощью режима НАБОР ДАННЫХ с клавиатуры дисплея. Наиболее удобны данные в ЭПЛК в табличной форме (рис. 3). Микросхема содержит 48 термов, представленных в таблице в виде строк. На экране монитора таблица представлена в виде четырех страниц. Данные в таблице могут находиться в следующих состояниях:

— в конъюнкцию включается прямое значение переменной (для матрицы И) или сигнал подается на вход установки RS-триггера (для матрицы ИЛИ), при этом в соответствующий разряд таблицы записывается символ 1;

— в конъюнкцию включается инверсное значение переменной или подается сигнал на вход сброса RS-триггера, при этом в таблицу записывается символ 0;

— переменная в конъюнкцию не включается или на входы триггера сигналы не подаются (записывается символ X);

— переменная не программируется (пишется символ —).

Перед программированием микросхемы пользователю часто необходимо убедиться в правильности набранных данных. В режиме ПРОГНОЗИРОВАНИЕ пользователь задает последовательность входных переменных. Программа моделирует работу будущего устройства в соответствии с таблицей данных и распечатывает на экране монитора вычисленные значения выходов. Программа также позволяет выявить ошибки и некорректности в таблице данных. Например, в регистре состояний указан код, по которому никогда не включается регистр обратной связи.

В режиме ПРОГРАММИРОВАНИЕ информация записывается в микросхему согласно таблице данных. В результате на экране монитора распечатывается сообщение «Микросхема запрограммирована» (в случае удачного выполнения режима) или «Брак» (в случае отказа при программировании).

В режиме ФУНКЦИОНАЛЬНЫЙ КОНТРОЛЬ проводится полная проверка запрограммированной микросхемы: на входы микросхемы последовательно подаются значения от 00...0 до 11...1 и информация с выходов микросхемы сравнивается с информацией, вычисленной в ЭВМ.

В режиме КОНТРОЛЬ ПОСЛЕДОВАТЕЛЬНОСТЕЙ пользователь задает последовательность входных комбинаций, подаваемых на микросхему. При этом на экране монитора распечатываются значения, полученные с выходов микросхемы.

В режимах ЧТЕНИЕ ФАЙЛА и ЗАПИСЬ ФАЙЛА данные программирования в виде отдельного файла записываются на ГМД или считываются с ГМД в ОЗУ ЭВМ (можно создать банки данных программирования).

Режим РАСПЕЧАТКА ТАБЛИЦЫ позволяет распечатать таблицу данных АЦПУ.

Таким образом, совокупность всех режимов работы подсистемы позволяет пользователю представить информацию программирования в удобном табличном виде, смоделировать работу будущего устройства, запрограммировать БИС ЭПЛК, провести проверку функционирования изготовленного устройст-

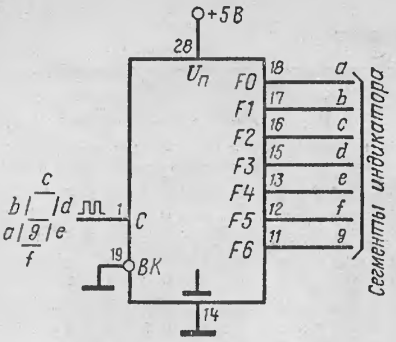


Рис. 4. Счетчик-дешифратор и 7-сегментный индикатор

ва, а также задокументировать полученные результаты работы.

В заключение рассмотрим простейший пример применения ЭПЛК как контроллера последовательностей. В таблице на рис. 3 записана программа реализации 4-разрядного счетчика-дешифратора семисегментного кода.

При подаче тактовой частоты на вход синхронизации триггеров на выходах F0...F6 последовательно появляются сигналы семисегментного кода символов 0...F. Последовательность включения определяется кодом регистра состояний. Так, если в каком-либо такте на регистре состояний установили код 000001, то в следующем такте включится комбинация с таким же кодом в триггерах обратной связи. Полученный счетчик-дешифратор, а также семисегментный индикатор показаны на рис. 4.

Основные преимущества подсистемы — небольшое время проектирования и удобство применения, так как все работы по проектированию, изготовлению и отладке устройств на базе ЭПЛК можно провести на одном рабочем месте.

Телефон 468-81-75, Москва

Статья поступила 24.12.87

УДК 681.326+681.3.06

В. М. Чурков, Л. П. Акумов

СИСТЕМА ПРОГРАММИРОВАНИЯ МИКРОСХЕМ, РАЗВИВАЕМАЯ ПОЛЬЗОВАТЕЛЕМ

Программирование микросхем — это завершающий этап изготовления устройств на их основе, который осуществляется пользователем с помощью программаторов. Для работы с программируемыми ПЗУ и программируемыми логическими интегральными схемами (ПЛИС) имеются программаторы двух классов: автономные и модульные. Автономный программатор — самостоятельный прибор, а модульным программатором управляет персональная ЭВМ (ПЭВМ). Номенклатура программируемых ПЗУ и ПЛИС программаторов ограничена их аппаратными возможностями, так как режимы программирования различных типов ПЗУ и ПЛИС, регламентируемые техническими условиями (ТУ), значительно отличаются друг от друга временными диаграммами. Как правило, любое расширение номенклатуры программируемых ПЗУ и ПЛИС изменяет аппаратуру программаторов. Разработчику часто приходится отказываться даже от оптимального в данной разработке типа программируемых ПЗУ и ПЛИС из-за отсутствия необходимых средств программирования.

Система программирования микросхем, развиваемая пользователем (СП), позволяет устранить указанные недостатки. СП (рис. 1) — это мощный программатор, управляемый ЭВМ («Электроника 60», «ДВК2М»). Такое сочетание дало СП большую гибкость и позволило пользователю производить ее настройку. В основе работы СП лежит формирование напряжений программирования в строго определенной последовательности на соединительной розетке (в соответствии с ТУ на микросхему и заданной оператором информацией). СП состоит из блока интерфейса (для связи с ПЭВМ), четырех блоков программирования (в каждом по 15 ячеек программирования), блока коммутации, узлов чтения и тест-контроля, соединительной розетки.

Первый тип блоков программирования обеспечивает на ячейках программирования блока выходные напряжения в пределах 0,2...37 В с шагом настройки 0,2 В и выходным током до 0,6 А; второй тип — 0,2...28 В с шагом настройки 0,2 В и выходным током до 1,3 А и третье — выходное состояние (высокий импеданс).

Блок коммутации адаптирует СП к программируемой микросхеме. В этом блоке пользователю дана возможность коммутации, регулировки фронта импульсов программирования, подключения дополнительных элементов (в соответствии с ТУ).

Узел чтения служит для проверки программируемой микросхемы на «чистоту» и правильность программирования.

Узел тест-контроля проверяет работоспособность аппаратуры, 88-контактная, трехрядовая соединительная розетка обеспечивает подключение к СП широкого ряда программируемых микросхем.

Посредством программных операций ПЭВМ осуществляются:

«Входной контроль» (проверка микросхемы на «чистоту»);

«Запись с эталона» (заполнение таблицы программирования с эталонной микросхемы);

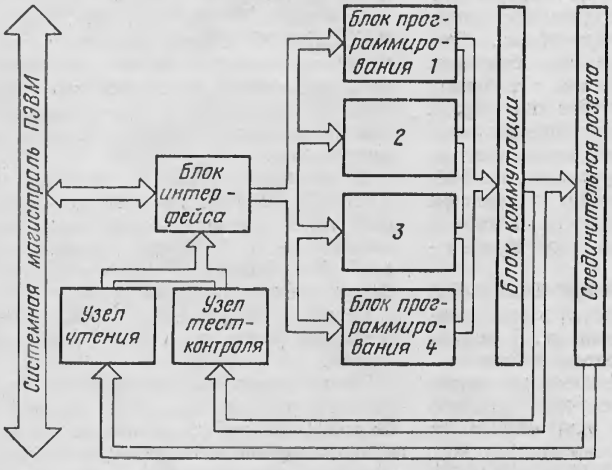


Рис. 1. Структура системы программирования микросхем, развиваемая пользователем

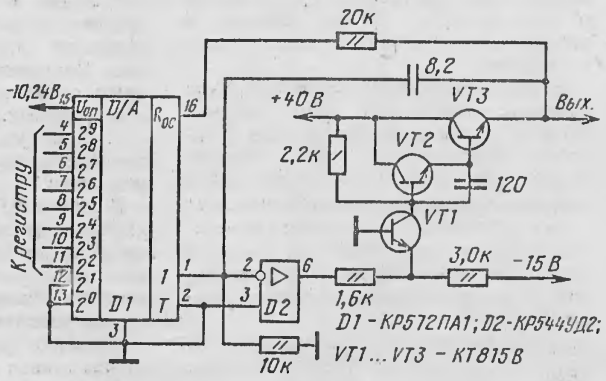
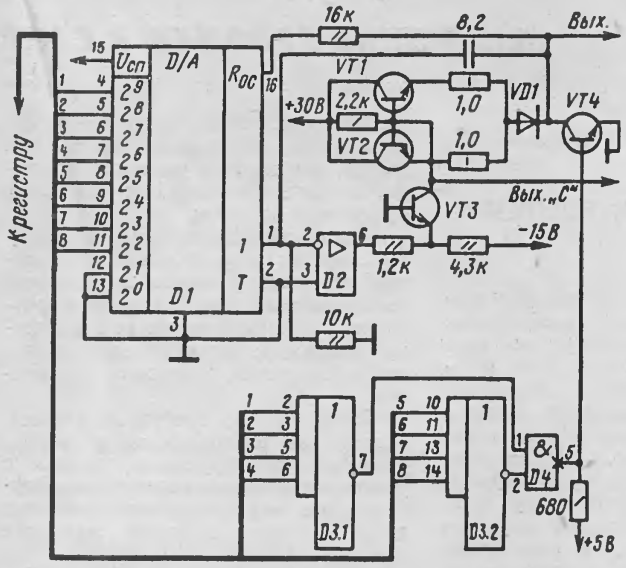


Рис. 2. Электрическая схема ячейки программирования первого типа



D1 - КР572ПА1; D2 - КР544УД2А; D3 - К155ЛЕ2; D4 - К155ЛИ5; VT1... VT3 - КТ829Г; VT4 - КТ815Б

Рис. 3. Электрическая схема ячейки программирования второго типа

«Сравнение с эталоном» (сравнение данных программирования с эталонной микросхемой);

«Набор данных» (заполнение таблицы программирования с терминала);

«Программирование» (программирование микросхемы);

«Контроль» (контроль запрограммированной микросхемы);

«Запись на диск» (запись данных таблицы программирования на гибкий магнитный диск);

«Вывод данных» (вывод таблицы программирования на печатающее устройство);

«Тест-контроль» (проверка работоспособности аппаратуры с выводом сведений о неисправностях на монитор).

Для программирования микросхемы, «имени» которой нет в пакете программ СП, пользователь, опираясь на техническое описание, руководство оператора, ТУ микросхемы, может сам построить временну́ю диаграмму программирования этой микросхемы. Для этого надо изменить данные в определенных адресах программы и присвоить «имя» полученному файлу. Заключительный этап подготовки СП — настройка блока коммутации, в котором осуществляется коммутация, установка фронта импульсов в соответствии с ТУ микросхемы. Пользователь проверяет построенную им временну́ю диаграмму и переходит к программированию.

Ячейки программирования (рис. 2, 3) — это программируемые источники напряжений, управляемые двоичным восьмиразрядным кодом с ПЭВМ. Код записывается по адресам регистров (на рис. 2 и рис. 3 регистры не показаны), жестко закрепленных за ячейками программирования. Пользователь, меняя в адресах ячеек первого типа код от 000_8 до 377_8, изменяет выходное напряжение в диапазоне 0,2...37 В.

Зависимость выходного уровня ячейки программирования второго типа от значения цифрового кода на входе D1 (рис. 3) выглядит так:

Значение цифрового кода на входе D1	Выходное значение ячейки программирования
000_8	Уровень цифровой земли
001_8	Высокий импеданс
$002_8...377_8$	0,4...28 В

Выход «С» соединен с блоком коммутации и предназначен для регулировки фронта импульса.

Телефон 468-81-75, Москва

Статья поступила 24.12.87

УВАЖАЕМАЯ РЕДАКЦИЯ!

В порядке оказания технической помощи прошу сообщить, согласовывали ли А. В. Коломиец и В. К. Горемыкин (авторы статьи в «МП».— 1987.— № 4.— С. 69) применение для защиты от перенапряжений ИС стабилитронов КС156А, КС162..., КС213 с МЭП СССР? По данным «Отраслевого центра по исследованию причин отказов ППП и ИС» (г. Мытищи) использование стабилитронов в режиме микротоков и ждущем режиме приводит к отказам стабилитронов.

г. Минск

На письмо читателя «МП» из г. Минска сообщаем следующее.

Авторы, публикуя приведенный материал, основывались на собственном многолетнем практическом опыте разработки, макетирования, отладки и испытания радиоэлектронных схем. При этом применение стабилитронов в указанном режиме всегда давало положительный результат. В самой статье подчеркнуто, что «...особенно при ремонте или настройке устройств на базе интегральных микросхем...» напряжение питания может превысить допустимое значение из-за «...ошибки при подключении, небрежности...». Работа же высоконадежных промышленных устройств всегда предполагает их эксплуатацию квалифицированным обслуживающим персоналом и стационарное питание их от блоков питания с высокоэффективными системами защиты. Это предполагает нецелесообразность использования рекомендуемой в статье защиты по цепям питания.

При макетировании разрабатываемых схем, отладке, испытании и ремонте рекомендуемая в статье защита по цепям питания представляется наиболее простой. К тому же измерения показывают, что, например, через стабилитрон КС156А при обратном напряжении 5 В протекает ток 0,55...1,2 мА (минимальный ток стабилизации для КС156А составляет 3 мА). Так что речь о режиме микротоков не идет. И еще, стабилитроны в режимах, аналогичных рекомендуемым в статье (т. е. с подводимым напряжением, меньшим, чем напряжение стабилизации), широко используются в различных ограничителях амплитуды, компараторах и т. п. (Например, см. книгу Алексенко А. Г., Коломбета Е. А., Стародуба Г. И. Применение прецизионных аналоговых микросхем.— М.: Радио и связь, 1985.— 177 С.).

А. В. Коломиец, В. К. Горемыкин

УДК 658.5.011.56

П. В. Вельтмандер, Г. В. Жуков, Л. В. Килина, А. М. Малярчук, В. Ф. Минин

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО КОНСТРУИРОВАНИЯ ДЛЯ ПЕРСОНАЛЬНОГО АРМ

Для автоматизации конструирования при существующем уровне развития средств вычислительной техники, методов и средств манипулирования данными целесообразна реализация интерактивных систем 2D-моделирования, ориентированных на пользователей-конструктов, работающих с деталировочными чертежами. Такие системы должны предоставлять конструктору интерактивные средства формирования и редактирования чертежей отдельных деталей и сборочных единиц, проведения типовых расчетов функциональных характеристик, расчетов на прочность и других, необходимых в повседневной работе, а также возможность получения сборочных чертежей из отдельных деталей с необходимыми средствами их контроля.

Рассмотренные соображения послужили основой для разработки систем автоматизации конструирования: МИК-С [1, 2] — монитор интерактивного конструирования, специализированный; МИК-М [3, 4] — монитор интерактивного конструирования, многоцелевой. Они разработаны для персонального автоматизированного рабочего места (АРМ) [5], позволяют формировать, редактировать и качественно перевыпускать комплекты текстовых и чертежных конструкторских документов.

В состав АРМ входят следующие технические средства: микроЭВМ «Электроника 60»; дисплей 15-ИЭ-00-013, оснащенный платой графического контроллера ГАММА-5 с разрешением 340×256 точек; графопостроитель ЭМ7042 с полем 300×420 мм; преддисплейный планшет Автограф-843 (может отсутствовать); электронный диск объемом 1 Мбайт; накопитель на гибких дисках Электроника ГМД-7012; накопитель на магнитной ленте СМ 5300.01; контроллеры связи с ЭВМ типов СМ4, ЕС1045, БЭСМ-6;

МИК-С и МИК-М работают на АРМе автономно. Реализованы в среде ОС РАФОС с Sj-монитором. При необходимости поддерживается связь с главными ЭВМ типов БЭСМ-6, СМ4, ЕС1045.

Организация диалога

Организация диалога пользователя одинакова для той и другой системы. Различия обусловлены специализацией МИК-С на класс осесимметричных деталей.

При ведении диалога на каждом шаге доступны директивы основного набора. После выбора пользователем требуемой директивы система выполняет запросы на ввод необходимых данных. Директива состоит из кода операции и списка параметров, который может быть пустым. Директива выбирается либо вводом ее имени с клавиатуры, либо указанием на преддисплейном планшете, где директивы изображены в виде пиктограмм.

Параметрами директивы могут быть числа, строки произвольной алфавитно-цифровой информации, координаты и указатели, идентифицирующие самостоятельный элемент изображения. Числа и произвольные строки вводятся с помощью алфавитно-цифровой клавиатуры терминала. Координатная информация, если она требуется, может вводиться с помощью одного из трех логических устройств класса локатор.

Локатор 1 реализуется с помощью преддисплейного планшета. Локатор 2 моделируется с помощью клавиш «перемещение символьного курсора». При нажатии на одну из четырех таких клавиш перекрестие перемещается по экрану в выбранном направлении. Скорость перемещения регулируется пользователем. Локатор 3 используется для точного задания координат в цифровой форме с клавиатуры. Все три устройства активируются одновременно, если по синтаксису команды требуется ввод координат. После начала манипуляции с каким-либо из устройств ввода координат система до завершения ввода очередной координаты деактивирует одно или два других. Одновременно активными могут быть локаторы 1 и 2. Устройство указания реализуется внешне как локатор 1 или 2, но результат ввода — указатель на выбранный самостоятельный элемент изображения.

Результат исполнения директивы, как правило, проявляется практически немедленно в визуальной форме. Если же исполнение директивы занимает заметное время, то пользователю выдаются информационные сообщения о том, чем занята система.

Система проектирования осесимметричных деталей МИК-С

Система МИК-С предоставляет пользователю следующие возможности:

построение чертежа отдельной детали: ввод эскиза детали, представляющего собой контур сечения детали до оси симметрии; назначение базовых поверхностей, от которых будут проставляться размеры; автоматическое построение полного вида детали; автоматическая простановка размеров от заданных базовых поверхностей; ввод требуемых условных обозначений (шероховатость, погрешность формы и так далее), допусков на указанные размеры, а также надписей и текстовых фрагментов чертежа;

редактирование чертежа: уже введенных эскизов; готовой детали с коррекцией указанных размеров, вызывающей изменение ее контура; результатов оразмеривания (иное размещение и построение размеров); дополнительных надписей, обозначений, текстовых фрагментов;

сборку узла из требуемых деталей, задание их расположения и формирование результирующего чертежа с удалением невидимых элементов и формирование результирующих размерных цепей с учётом допусков отдельных деталей;

документирование на требуемые устройства;

выполнение архивных работ.

Построение чертежа отдельной детали разбивается на ряд уточнений ее математической модели. Первоначально строится упрощенная модель, не учитывающая отверстия, пазы, резьбы и другие конструктивные элементы. Следующий шаг — введение дополнительных конструктивных элементов (пазов, отверстий, резьб). После этого автоматически строится полный вид детали. Генерируется и оптимизируется совокупность размерных цепей, которая дополняет математическую модель детали.

Полученная после этого модель может дополняться атрибутами конструкторско-технологического чертежа (шероховатость, погрешность формы и расположения, клеймение, полки-выноски с технологическими текстами и т. д.), называемыми условными обозначениями.

По окончательной модели чертежа формируются математические модели видов справа, слева, местных видов (изображений определенных участков детали в заданном месте).

После получения законченных моделей отдельных деталей на их основе производится компоновка и обсчет сборочных узлов и выпуск сборочных чертежей (рис. 1).

При вводе координатной информации используется понятие текущей точки, т. е. любые построения могут производиться от конечной точки предыдущего построения; при вводе параметров директив — принцип умолчания. Значения по умолчанию выбирались наиболее соответствующими обычной конструкторской практике.

На этапе построения чертежа пользователю предоставлены естественные для конструктора элементарные построения в виде точек, отрезков вертикальных, горизонтальных и наклонных прямых, дуг окружности, ломаных, сопряжений, фасок, вертикальных и горизонтальных замыканий — линий от текущей точки до оси детали и до на-

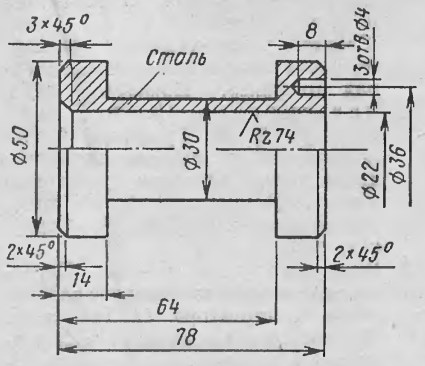


Рис. 1. Полученный чертеж детали

чала детали вдоль ее оси, соответственно диаметральных линий, отверстий, резьб и т. д.

Пользователь работает в системе координат, привязанной к осевой линии детали и левой торцевой поверхности ее с миллиметровой сеткой измерения. Для управления простановкой размеров может быть указано необходимое число базовых поверхностей. При вводе условных обозначений предусмотрены директивы расчета толщины от указанной на поверхности точки до внутренней поверхности; расчета размера сечения детали в указанном месте; построения углового размера; задания для поверхности шероховатости, маркирования, клеймения и погрешности формы (биения, параллельности и т. д.).

Математическая модель объекта проектирования, содержащая геометрические, конструктивные и атрибутные данные, представляется в виде прямоугольной матрицы. Чтение, занесение и (или) модификация данных выполняются специальными интерфейсными модулями. Такая организация структуры данных, несмотря на определенное увеличение ее объема, обеспечивает простой доступ к данным, легкое продолжение структуры на внешнюю память и сравнительно несложный интерфейс с программами выполнения проектировочных расчетов.

Система МИК-С реализована в виде набора автоматически вызывающих друг друга задач. Связь между отдельными задачами осуществляется с помощью файлов на внешней памяти (электронном диске). Интерфейсные модули работы с графическими устройствами написаны на языке MACRO, остальные программы системы — на Фортране-4. Объем программного кода составляет около 750 Кслов.

Основная область применения системы: оперативная подготовка, сохранение в архиве и выпуск графических и текстовых компонентов документации, удовлетворяющих требованиям ЕСКД при существенном сокращении сроков получения чертежей. Время обучения работе с системой 1—2 дня.

Система автоматизации конструирования произвольных деталей МИК-М

Система МИК-М реализует этап формирования, оперативного редактирования и перевыпуска комплекта документации для произвольных деталей. Процесс конструирования в этом случае заключается в построении геометрической модели плоского объекта (чертежа), задании размеров, дополнительных обозначений и надписей.

Директивы основного набора системы МИК-М:

управление системой — отмена и повторение директивы, задание зоны чувствительности перекрестия, установление и отмена вспомогательного позиционирующего растра, копирование изображения на требуемое устройство (лента, диск, графопостроитель), манипулирование архивами и др.;

формирование изображений из примитивных геометрических и конструктивных элементов (точки, отрезки, ломаные, дуги окружностей, скругления, фаски, затупления, штриховки, прямо- и многоугольники, скругленные прямоугольники, атрибуты примитивных элементов и т. д.);

выполнение дополнительных построений — нанесение линейных, угловых и свободных размеров, допусков, условных обозначений и др.;

манипулирование изображением — установление окна просмотра, аффинные преобразования изображения, построение объемных каркасных изображений вращением выполненного контура, назначение системы координат и др.;

ввод координат и указание объекта на изображении.

Все графические элементы (точки, отрезки, дуги, окружности) могут быть индивидуально доступны для манипуляций (например, указания, стирания) и организованы в графические объекты, над которыми можно осуществлять действия как над самостоятельными, например аффинные преобразования.

В системе МИК-М перекрестие имеет определенную зону захвата, так что при задании координат или указании графические элементы, попадающие в зону захвата перекрестия, обладают эффектом притяжения и изменяют (уточняют) результаты ввода. Это позволяет компенсировать главный недостаток преддисплейного планшета — невысокую точность задания координат. Для правильного ввода чертежа детали достаточно ввести с клавиатуры несколько точных значений координат точек или других геометрических элементов, а все остальные построения легко осуществляются с помощью планшета, что значительно удобнее и быстрее.

В системе нет средств автоматического нанесения размеров, все размеры задает пользователь. Для этого ему необходимо задать требуемый тип размера (линейный, угловой, радиальный), указать элемент, для которого

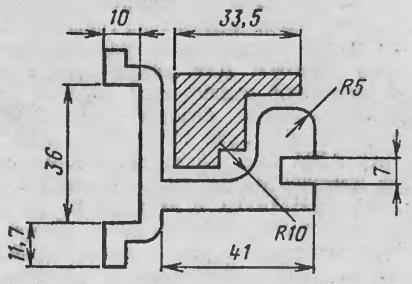


Рис. 2. Чертеж плоской детали

задается размер, и его примерное расположение.

Редактирование графической информации — указание стираемых и ввод новых геометрических элементов или объектов. При отмене штриховки последняя из введенных удаляется для указанного контура (рис. 2).

Система располагает средствами построения каркасных изображений 3D-объектов. Каркасные тела либо получаются вращением контура вокруг оси, либо задаются кусками трехмерных поверхностей вида Y=F(X, Z) для заданных диапазонов значений X и Z. Дискретность сетки каркасных изображений указывается пользователем.

В системе также реализован широкий набор функций по подготовке и выводу текстовых конструкторских документов, позволяющих ориентировать, масштабировать, позиционировать на чертеже и листовать тексты, содержащие как обычные алфавитно-цифровые строчные и прописные, латинские и русские символы, так и специальные, определенные стандартами ЕСКД.

Рассмотренные средства позволяют формировать, редактировать и выпускать документацию, автоматизировать проектирование технологических процессов механообработки.

Выводы

Опыт эксплуатации данных систем автоматизации конструкторских работ позволяет сделать следующие выводы.

1. Предоставление конструкторам только графических средств без проведения каких-либо расчетов значительно снижает их заинтересованность в использовании средств автоматизации. Предоставление же интерактивных средств проведения трудоемких расчетов различных характеристик, а также автоматизация таких операций, как контроль сборочных узлов на совместимость, расчет размерных цепей сборочных узлов с учетом допусков и посадок и прочее, побуждают конструкторов изучать и применять новые средства в качестве рабочего инструмента.

2. Актуальным остается вопрос о технических средствах АРМа. Основные ограничения, тормозящие дальнейшее развитие,— малая вычислительная мощность микроЭВМ, ограниченность

объема ее ОЗУ и быстрой внешней памяти, малое разрешение графического дисплея, надежность графопостроителя ЭМ7042. Для ликвидации этих ограничений планируется перейти на более мощную базовую микроЭВМ типа МС1212 либо КРОНОС.и оснастить АРМ быстрой внешней памятью на жестких дисках, графическим дисплеем ГАММА-7.1 с экраном 51 см и разрешением 1024×768 точек. Открытым остается вопрос о комплектации АРМа более совершенным графопостроителем и качественным печатающим устройством.

Телефон 32-47-51, Новосибирск

ЛИТЕРАТУРА

1. Вельтмандер П. В., Жуков Г. В., Кочерыгин И. А. и др. Автоматизированная система конструирования осесимметричных деталей // Тез. докл. Всесоюз. школы-семинара «Информатика и интерактивная компьютерная графика».— Цахкадзор, 1987.
2. Вельтмандер П. В., Жуков Г. В., Килина Л. В., Малярчук А. М. Организация графического диалога в системе автоматизированного проектирования осесимметричных деталей // Материалы IV Всесоюз. конф. по проблемам машинной графики.— Серпухов, 1987.
3. Вельтмандер П. В., Малярчук А. М. Система автоматизации конструирования, основанная на подходе чертежей // Тез. докл. Всесоюз. школы-семинара «Информатика и интерактивная компьютерная графика».— Цахкадзор, 1987.
4. Вельтмандер П. В., Малярчук А. М. Графический диалоговый язык системы автоматизации конструирования // Материалы IV Всесоюз. конф. по проблемам машинной графики.— Серпухов, 1987.
5. Вельтмандер П. В. Персональное автоматизированное рабочее место конструктора // Тез. докл. Всесоюз. школы-семинара «Информатика и интерактивная компьютерная графика».— Цахкадзор, 1987.

Статья поступила 05.05.88

УДК 621.396.6.001.24

И. Г. Выскребцов, Д. А. Журавский, А. Я. Олейников, Н. В. Толстоброва, А. В. Элбакидзе

СИСТЕМА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ И ИЗГОТОВЛЕНИЯ ПЕЧАТНЫХ ПЛАТ НА БАЗЕ МИКРОЭВМ

При машинном проектировании печатных плат с максимально допустимой стандартами плотностью компоновки элементов за ограниченное время трассируется 90...95 % соединений [1]. Дальнейшее продолжение трассировки на ЭВМ приводит к резкому возрастанию затрачиваемого машинного времени и экономически не оправданно.

При разработке систем автоматизации экспериментов (САЭ) в области радиофизики на базе микроЭВМ «Электроника 60» и СМ ЭВМ с применением КАМАК аппаратура сопряжения комплектуется готовыми универсальными модулями КАМАК. Однако при создании каждой следующей системы необходимо разрабатывать новые модули со специальными функциями для решения задач конкретного пользователя. Их изготовление силами КБ занимает 1—2 года, что неприемлемо.

Для создания САПР макетных плат можно использовать системы автоматизации экспериментов при условии обеспечения их необходимыми техническими и программными средствами.

В последнее время широкое распространение в САЭ получили растровые графические дисплеи на базе бытовых цветных телевизоров или видеомониторов, управляемые несложными модулями-контроллерами [2, 3], в стандарте КАМАК или «Электроника 60», с высокой надежностью и низкой стоимостью по сравнению с векторными графическими дисплеями. Применение такого дисплея и внешней полупроводниковой памяти типа «электронного диска» [4, 5] с микроЭВМ ДВК, «Электроника 60» или мини-ЭВМ типа СМ позволяет использовать описываемую САПР без существенных доработок.

Комплекс дает возможность выполнять следующие этапы: интерактивное размещение элементов на поле печатной платы; автоматическую трассировку соединений с отображением хода процесса на экране цветного видеомонитора; графическое отображение результатов размещения и трассировки на экране цветного видеомонитора или на графопостроителе; редактирование результатов автоматической трассировки с отображением рисунка печатной платы на экране цветного видеомонитора; получение управляющей перфоленты для фотопостроителя и сверлильного станка: изготовление фотошаблонов и сверление отверстий непосредственно под управлением ЭВМ при использовании станков для производства печатных плат (например, АДМАП производства ВНР).

Состав технических средств комплекса: процессор с системой команд типа СМ3 («Электроника 60», ДВК1, ДВК2, ДВК3; СМ3, СМ4, СМ1300, СМ1420 и др.), ОЗУ объемом 28 Кслов, алфавитно-цифровой дисплей 15ИЭ-00-013, накопитель на дисках ГМД-7012, перфоратор для вывода управляющих перфолент, крейт КАМАК с модулями вывода на цветной телевизор, ОЗУ в качестве электронного диска, модуль управления графопостроителем.

Состав прикладного ПО:

программа интерактивного размещения редактирует установочные параметры (координаты, ориентацию) элементов печатной платы в файле связей, работает в диалоговом режиме. Файл связей создается любым редактором текста в специальном формате, содержит размеры платы и описание элементов: схемное обозначение, типы, координаты установки, ориентацию, перечисления соответствий номеров выводов и названий цепей;

программа машинной трассировки проводников обеспечивает автоматическую трассировку по составленному файлу связей с библиотекой и файлом начального конструктива. В программе используется волновой алгоритм, внешняя память для временно́го файла (256 блоков) под модель печатной платы на устройстве с произвольным доступом («электронный диск» [4, 5]). В конце программы выводится сообщение о числе неразведенных проводников, контактных площадках, переходных отверстиях и их общем числе. Возможно принудительное завершение работы программы;

выходной файл конструктива вводится в программу редактирования графической информации. С помощью этой программы доводятся неразведенные проводники и заканчивается процесс трассировки или вносятся изменения и результаты работы рассматриваются в качестве входного файла программы трассировки. Основные команды перечислены на экране алфавитно-цифрового дисплея в виде меню;

цоколевочная библиотека (текстовый файл спецформата) состоит из каталога и описаний элементов, использует язык Макроассемблера. Ее элементы кодируются с помощью специальных макроопределений.

К настоящему времени с помощью описываемой САПР спроектировано более 100 печатных плат размерами от 138×240 до 275×240 мм и изготовлены их фотошаблоны. Набранная статистика позволяет привести некоторые усредненные параметры проектирования: при проектировании платы размерами 138×240 мм, содержащей 45...50 микросхем, 30 резисторов, 20 конденсаторов, время, затраченное на автоматическую трассировку,— около 1 ч., число разведенных проводников после автоматической трас-

сировки — 95 %, время, затраченное на редактирование (доведение неразведенных связей),— до 3 ч., время вывода фотошаблонов двух сторон платы — до 4 ч.

Телефон 931-21-00, Москва

ЛИТЕРАТУРА

1. Белов А. Н., Выскребцов И. Г., Журавский Д. А., Толстоброва Н. В., Элбакидзе А. В. Автоматизированная система конструкторского проектирования печатных плат.— Препринт.— М., 1987.— 463 с.— (ОИЯИ, № 4).
2. Якушев В. С. Цветной телевизионный КАМАК-дисплей // Автометрия.— 1984.— № 4.— С. 83—96.
3. Семенов Ю. Б., Челноков Л. П., Портие Р. Драйвер цветного телевизора.— Препринт.— М., 1981.— 271 с.— (ОИЯИ, № 13).
4. Сорокин Ю. Ю., Лаврентьев В. В., Максимяк С. П., Субач В. В. «Электронный диск» для микроЭВМ «Электроника 60М» // Микропроцессорные средства и системы.— 1986.— № 5.— С. 82.
5. 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 04.08.87

УДК 681.3.06

К. П. Голиков, И. Е. Педанов, В. Н. Щурий

ПОДГОТОВКА ИНЖЕНЕРНОЙ ДОКУМЕНТАЦИИ С ПОМОЩЬЮ СИСТЕМЫ РЕДГРАФ

Система РЕДГРАФ, разработанная в ВЦ АН СССР [1—3],— универсальный интерактивный графический редактор для автоматизации подготовки графической документации на различных этапах проработки инженерных изделий: при создании эскизных проектов, деталировочных и сборочных чертежей, подготовке технологической документации.

Функционально РЕДГРАФ обеспечивает ввод и редактирование в режиме диалога графической и текстовой информации, хранение ее в архиве на внешних носителях, преобразование с целью получения новых графических документов и вывод на экраны дисплеев для просмотра или на устройства получения твердых копий (графопостроители, принтеры).

На основе графических файлов в РЕДГРАФе реализован выход на систему подготовки программ для станков с ЧПУ (система САП*СМ), что позволяет осуществить сквозную автоматизацию инженерных работ от проектирования до производства.

Как программный продукт РЕДГРАФ — развитый пакет программ на Фортране со встроенным диалоговым монитором, что обеспечивает достаточную мобильность системы при переносе ее на другие ЭВМ и в другие операционные среды, а также расширение функциональных возможностей системы.

Для выполнения функций базовой графики используется универсальная графическая система на основе GKS, разработанная в секторе машинной геометрии и графики ВЦ АН СССР [4].

Графическая станция реализована на базе персонального компьютера типа IBM PC XT/AT (ОС MS DOS) и совместимых с ним. В состав рабочего места инженера входят также графический дисплей или достаточно мощный графический контроллер с монитором высокого разрешения (1024×780 точек растра), кодировочный планшет настольного типа и графопостроитель. Не вызывает сомнения и использование в качестве базовой ЭВМ на рабочей станции микроЭВМ ЕС1841, так как имеется опыт работы с системой на ЕС1839. Существенное требование — наличие жесткого диска емкостью не менее 5 Мбайт. Возможна эксплуатация системы РЕДГРАФ и на станциях с минимальными возможностями: один дисплей с адаптером типа CGA, клавиатура и графопостроитель. Однако следует отметить, что даже при наличии более совершенного адаптера типа EGA производительность и комфортность работы значительно снижаются.

Основные приемы работы в системе РЕДГРАФ

Работа пользователя в системе выполняется с помощью меню команд РЕДГРАФа (рис. 1). Команды вводятся либо указанием соответствующей кнопки (поля) меню, закрепленного в любом месте цифрового планшета, либо в числовом виде с клавиатуры ЭВМ.

Процесс работы по формированию чертежа ведется с использованием как бы двух наложенных один на другой чертежных листов одинакового размера — чернового и чистового (в системе они соответственно называются задним и передним планами). На экране элементы заднего плана отображаются с пониженной яркостью. Ввод и различные преобразования отдельных элементов или частей чертежа производятся на черновике. После того как данная часть чертежа готова, она помещается на задний план.

Разделение на задний и передний планы играет существенную роль при построении и редактировании чертежа. В частности, разделение на два плана обеспечивает наглядные и удобные средства нейтрализации ошибок пользователя. Если какое-то построение оказалось неудачным, достаточно просто очистить передний план.

Процесс формирования чертежа во многом напоминает работу за кульманом. Вначале пользователь командой ввода чисел задает размеры чертежного листа в миллиметрах по осям x и y, а затем командой «лист» устанавливает выбранный формат. Начало отсчета при этом помещается в левый нижний угол листа. При необходимости можно переходить в локальную систему координат с началом в указанной точке чертежного листа. Кроме того, предусмотрена работа в заданном масштабе. При этом все задаваемые в дальнейшем параметры геометрических объектов автоматически пересчитываются в соответствии с установленным масштабом.

После выполнения описанных операций можно приступать к построению чертежа с помощью команд ввода отдельных элементов: отрезков, дуг, окружностей и т. п. Каждая команда требует предварительного задания геометрических параметров элемента. Например, для построения отрезка нужно задать две точки, для построения окружности — положение центра и радиус и т. д. Пользователь может задать вид линии и цвет.

В системе РЕДГРАФ точки вводятся несколькими способами. Командой «координаты» задают точку числовым координатным выражением. Причем предусмотрен ввод декартовых и полярных координат точки, а также координатного выражения, определяющего координаты точки по оси, направленной под заданным углом к горизонтали и перпендикулярно к ней. Командой «приращение» задаются координаты точки относительно другой, ранее определенной точки чертежа. С помощью локатора (подвижного крестообразного указателя, управляемого с планшета или с клавиатуры) можно задать точку на экране, выбрать произвольную точку на листе или указать уже существующую (например, граничная точка отрезка или дуги). Необходимые для построения чертежа точки можно получить с помощью команд координатного расчета, например, как середину указанного отрезка, точку пересечения двух отрезков, точку касания к данной окружности и т. п.

Особенно следует отметить возможность задания точек с использованием дискретной сетки, точка привязки и шаг

* Система автоматизированной подготовки управляющих программ для станков с ЧПУ

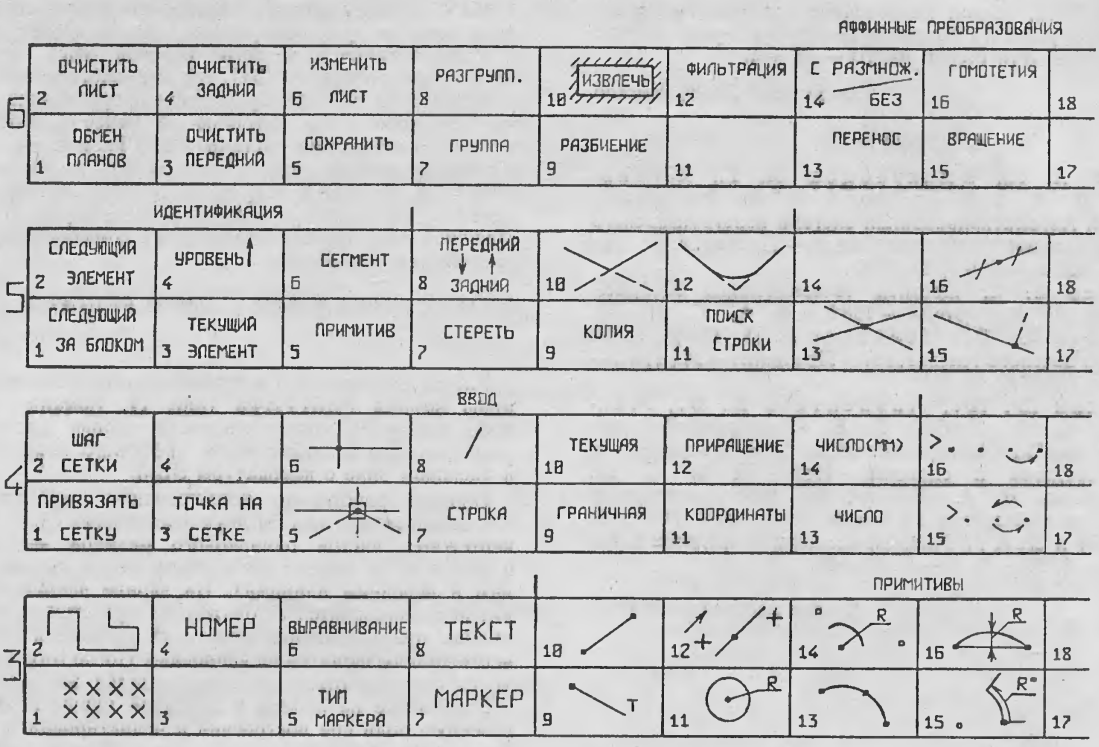


Рис. 1. Фрагмент меню системы РЕДГРАФ

которой задаются самим пользователем. На любом этапе работы изображение сетки можно показать на экране дисплея с помощью точечного растра. При работе на сетке координаты указанной локатором точки автоматически округляются до координат ближайшего узла сетки.

Отрезок прямой можно построить 10 различными способами. Например, отрезок между двумя точками строится после задания этих точек любой из перечисленных выше команд (рис. 2). В системе возможно построение совокупности связанных отрезков — ломаной, точки которой располагаются в узлах дискретной сетки и вводятся на экране непосредственным указанием их локатором.

Для построения окружности пользователю необходимо задать положение центра и ввести число, определяющее радиус.

Дуга строится четырьмя способами: по трем точкам (начальной, конечной и промежуточной); по заданному радиусу, центру и проекциям двух точек на несущую окружность; по радиусу, центру, начальной точке и угловому размеру (в градусах); по начальной и конечной точкам и стреле прогиба (см. рис. 2).

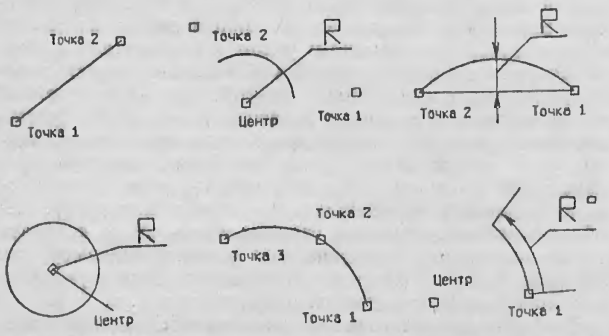


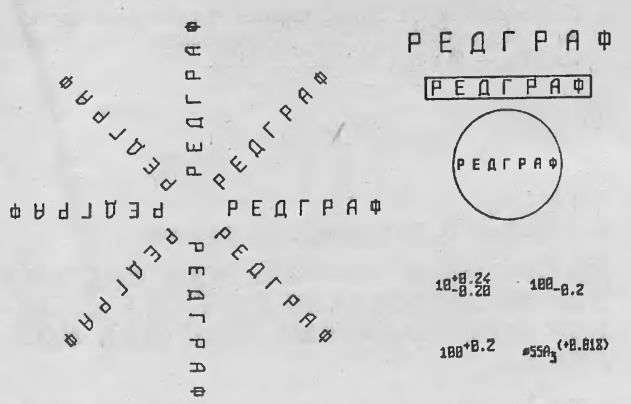
Рис. 2. Построение основных элементов в системе РЕДГРАФ

Текстовая информация в системе РЕДГРАФ вводится в виде строк символов (рис. 3). Высота символов, отношение ширины к высоте, наклон строк относительно горизонтали устанавливаются соответствующими командами. В системе имеется четыре набора символов: русский, латинский, специальный и КОИ-7. Для записи строки текста на чертеже пользователь предварительно вводит ее с клавиатуры, а затем указывает локатором, где она должна располагаться.

Соответствующими командами устанавливают режимы выравнивания строк, обрамления текста (строка помещается в прямоугольную или круглую рамку).

Работа конструктора с чертежом требует внесения многочисленных дополнений и исправлений, т. е. редактирования полученного графического изображения. На этом этапе работы пользователю предоставляются самые разнообразные возможности. Основная операция — идентификация (указание) того элемента или совокупности элементов, которые нужно изменить. Идентификация любого элемента чертежа производится с помощью локатора — установкой указателя (обычно специального подвижного маркера) на соответствующий объект. При успешном завершении операции пользователю выдается сообщение о типе элемента, плане его принадлежности, виде линии и цвете. Только после того, как пользователь убедится, что элемент выбран правильно, он может производить с ним все необходимые операции: удалять, переносить с заднего плана на передний или в другое место чертежа, менять вид линии или цвет, копировать и т. п.

Одним из приемов, значительно повышающих эффективность работы, является объединение отдельных элементов в достаточно крупные блоки. Элементы объединяются в соответствии с критериями, характеризующими отдельные законченные части данного чертежа: изображение отдельной детали, узла или целой сборки, какое-либо стандартное обозначение на чертеже. В системе различают два вида блоков — группы и фрагменты. Группа формируется из элементов, предварительно вынесенных на передний план.


РЕДГРАФ
РЕДГРАФ
РЕДГРАФ

ление и сортировка упорядоченного контура, натяжение контура на каркас опорных элементов, удаление петель на эквидистанте.

Размеры на чертеже проставляются в системе РЕДГРАФ в полуавтоматическом режиме. Выносные и размерные линии строятся автоматически по предварительно заданным точкам, после чего пользователь указывает локатором место размещения размерной надписи. Предусмотрена простановка линейных размеров и размеров радиусов, диаметров, дуг, углов. Можно выбрать один из нескольких вариантов размерной цепи, а также указать наличие или отсутствие стрелок на концах размерной линии. Размерная надпись может включать специальные обозначения и допуски. Штриховка выполняется автоматически с заданным шагом и углом наклона к горизонтали.

Любой чертеж, созданный пользователем, можно записать в архив чертежей в виде отдельного файла (называемого кадром) с уникальным именем и вызвать для просмотра или редактирования. Специальными командами просматривается каталог архива, запрашивается информация о текущем кадре, удаляется кадр.

Любой из архивных кадров можно вызвать в качестве фрагмента на другой чертеж, поэтому в архиве хранятся не только законченные чертежи, но и кадры, являющиеся составными частями других чертежей. По мере наполнения архива производительность труда еще больше возрастает, так как каждая проектная группа формирует свой специфический архив наиболее часто употребляемых графических изображений и символов.

Последовательности команд системы, которые пользователю приходится выполнять многократно, объединяются в наборы, называемые макрокомандами (макро). Определение макро — это просто список команд, вызываемый указанием одной кнопки меню. Определение начинается командой «открыть макро». Затем последовательно нажимаются те кнопки меню, которые пользователь хочет записать в макро. Макрокоманды можно сохранять на диске и считывать в следующем сеансе работы.

Твердые копии подготовленной чертежно-конструкторской документации получаются с использованием промежуточного файла, формируемого системой. Формат файла выбран из соображений его наибольшей мобильности при переносе на другие ЭВМ. Выводится файл на различные графопостроители специальной программой, модифицируемой с учетом формата команд нового устройства вывода.

Первоначально РЕДГРАФ был реализован на ЭВМ «Электроника 100/25» с использованием аппаратных средств СМП «Кулон» (в ОС РАФОС). Перенос системы на микроЭВМ типа IBM PC XT/AT значительно расширил сферу внедрения и применения системы. Из отечественных микроЭВМ система РЕДГРАФ успешно работает на ЭВМ «Электроника 85» в Институте проблем кибернетики АН СССР. В настоящее время система внедрена и успешно используется на десятках предприятий не только для подготовки инженерно-конструкторской документации в машиностроении, но и для разработки и редактирования различной графической информации схемотехнического характера: принципиальных схем, печатных плат, схем автоматики, гидравлики и т. п.

Телефон 135-90-55, Москва

ЛИТЕРАТУРА

1. Голиков К. П., Педанов И. Е. РЕДГРАФ — система редактирования чертежной документации. Основные функции.— М.: ВЦ АН СССР, 1981.— 64 с.
2. Голиков К. П., Педанов И. Е. РЕДГРАФ: интерактивная система подготовки чертежно-конструкторской документации // Тез. докл. II Всесоюзной конференции «Методы и средства обработки сложной графической информации».— Горький, 1985.
3. Щурий В. Н. Новые возможности системы РЕДГРАФ // Сб. «Программное обеспечение САПР».— М.: ВЦ АН СССР, 1987.
4. Страхов А. В. Универсальная графическая система на основе GKS // Сб. «Программное обеспечение САПР».— М.: ВЦ АН СССР, 1987.

Статья поступила 05.07.88

УДК 681.3.06:54

В. А. Бобков, О. С. Кислюк, Т. А. Боровина

ДИАЛОГОВАЯ ГРАФИЧЕСКАЯ СИСТЕМА ДЛЯ ИССЛЕДОВАНИЯ СТРУКТУР СЛОЖНЫХ МОЛЕКУЛЯРНЫХ ОБЪЕКТОВ МОЛЛЮСК-2

Изучение пространственного строения моделей молекулярных объектов — одна из важных сфер применения машинной графики [1—3]. В настоящее время возник даже специальный термин — молекулярная графика.

Наиболее интересны случаи применения систем молекулярной графики, когда ЭВМ используется как инструмент графического вывода и численных расчетов [4, 5] или при сочетании визуального анализа молекулярных структур и анализа этих структур как комбинаторных объектов (например, изучение зависимости структура-свойство).

При создании предлагаемой системы учитывались анализ предметной области и опыт использования системы МОЛЛЮСК-1 [6]. Основные требования, предъявляемые к системе молекулярной графики:

1. Широкий выбор типов моделей и операций над ними, позволяющий дать взаимодополняющую информацию о различных свойствах молекулярных объектов. Так, шаростержневая модель отражает внутреннее строение молекулы, а Ван-дер-Ваальсова поверхность, раскрашенная в соответствии с интенсивностью поля,— химическую активность.

При работе с моделями часто возникает потребность упрощать и выделять их части, удовлетворяющие некоторому условию, и т. д. Например, конкретный запрос, возникающий при эксплуатации системы МОЛЛЮСК-1: построить изображение модели белка, состоящее из триптофанов (название аминокислоты), имеющих выход на поверхность белка, совместно с цепочкой аминокислот в виде последовательности шаров, соединенных стержнями. Пример демонстрирует, что желательно иметь широкий набор операций над моделями и специальный язык для формирования сложных запросов.

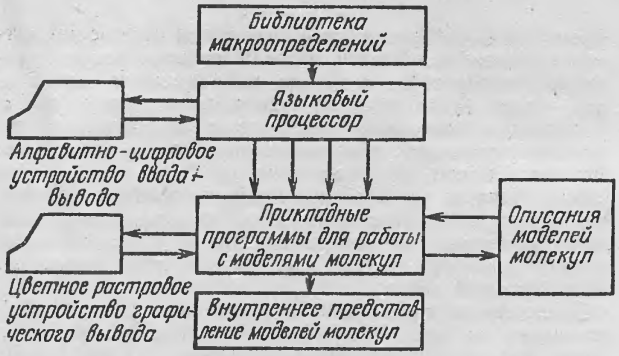
2. Возможности системы в построении цветных полутоновых изображений пространственных молекулярных объектов: получение изображений моделей, состоящих из нескольких тысяч атомов, за время, не превышающее несколько десятков секунд; высокая реалистичность изображений (тени, уменьшение освещенности с глубиной, полупрозрачность и т. д.); построение изображений специального вида и комбинированных (окраска объекта в соответствии с интенсивностью электростатического поля, полупрозрачное наложение двух моделей и т. д.).

3. Изучение особенностей пространственного строения сложных молекул (в частности, биополимеров), которые состоят из одной или нескольких цепочек. Такие цепочки (для белков, например, это цепочки аминокислот) можно исследовать, применяя технику сопоставления с образом для строк символов, дополненную учетом пространственного строения биополимера.

4. Геометрические преобразования, обеспечивающие наложение моделей молекул при изучении их сходства и сближение моделей при изучении взаимодействия двух молекул.

Кроме того, система допускает подключение прикладных программ пользователя для вычисления свободной энергии, интенсивности полей и т. д.

Общение пользователя с системой (см. рисунок) осуществляется на специальном языке. Текст языка транс-



Архитектура системы молекулярной графики

лируется в последовательность вызовов прикладных программ для работы с моделями молекулярных объектов и программ пользователя. С помощью прикладных программ, вызываемых языковым процессором, можно ввести описание структуры молекулы в одном из внешних форматов. Для представления молекулы используются следующие типы моделей: шаро-стержневая; из Ван-дер-Ваальсовых шаров; модель, в которой каждому атому соответствует эллипсоид; кристаллическая решетка в виде совокупности многогранников; поверхность, доступная растворителю.

Язык взаимодействия с системой. Типы данных — целые, вещественные, строковые, молекулярные объекты. Молекулярные объекты — это упорядоченные последовательности атомов. Каждый атом снабжен списком атрибутов: координаты, радиус, цвет, тип атома, тип структурной единицы (для белков — тип аминокислоты) и др. Так как структурная единица представлена в списке атомов в виде непрерывного сегмента, имеет место упорядочение структурных единиц (для белков молекулярный объект — упорядоченная последовательность аминокислот).

При шаро-стержневой модели над атомами молекулярного объекта строятся связи, каждая из которых ссылается на два связываемых атома. При использовании модели, состоящей из многогранников, над атомами молекулярного объекта строятся многоугольники.

Всем переменным в языке приписан тип. Он определяется первой буквой имени переменной: I — целая, R — вещественная, C — строковая, S — молекулярный объект. Любое обращение к системе есть последовательность операторов, а исполняемой единицей языка является оператор.

В языке существуют четыре типа операторов: вызов прикладной программы, IF, NIF, DO. При выполнении каждого оператора вырабатывается логическое значение («истина» или «ложь»). Кроме того, операторы могут изменять значение специальной переменной — курсора, указывающего на некоторый атом в списке атомов молекулярного объекта.

Оператор вызова прикладной программы имеет вид ⟨имя⟩ (⟨список АРГУМЕНТОВ⟩). Аргументы — переменные или константы. При исполнении оператора вызывается прикладная программа, которая вырабатывает логическое значение, может изменять значения курсора и аргументов.

Операторы IF, NIF и DO имеют вид:

IF ⟨ ЛЕВАЯ ЧАСТЬ ⟩ : ⟨ ПРАВАЯ ЧАСТЬ ⟩ FI
NIF ⟨ ЛЕВАЯ ЧАСТЬ ⟩ : ⟨ ПРАВАЯ ЧАСТЬ ⟩ FIN
DO ⟨ ЛЕВАЯ ЧАСТЬ ⟩ : ⟨ ПРАВАЯ ЧАСТЬ ⟩ OD.

Правая часть есть последовательность операторов, разделенных «;», она выполняется только в том случае, если выполнена левая часть. Вид левой части определяется правилом

⟨ЛЕВАЯ ЧАСТЬ⟩ ::= ⟨ОПЕРАТОР⟩
| ⌝⟨ЛЕВАЯ ЧАСТЬ⟩
| ⨀⟨ЛЕВАЯ ЧАСТЬ⟩
| ⟨ЛЕВАЯ ЧАСТЬ⟩&⟨ЛЕВАЯ ЧАСТЬ⟩
| ⟨ЛЕВАЯ ЧАСТЬ⟩ ! ⟨ЛЕВАЯ ЧАСТЬ⟩
| (⟨ЛЕВАЯ ЧАСТЬ⟩.

Левая часть считается выполненной, если она:

есть оператор, который выработал логическое значение «истина»;

имеет вид:

⟨ЛЕВАЯ ЧАСТЬ 1⟩ ! ⟨ЛЕВАЯ ЧАСТЬ 2⟩ и выполнена хотя бы одна из левых частей, если выполнены обе, то результирующая левая часть будет выполнена дважды;

⟨ЛЕВАЯ ЧАСТЬ 1⟩ & ⟨ЛЕВАЯ ЧАСТЬ 2⟩ и выполнены ⟨ЛЕВАЯ ЧАСТЬ 1⟩ и ⟨ЛЕВАЯ ЧАСТЬ 2⟩;

⌝⟨ЛЕВАЯ ЧАСТЬ 1⟩ и если не выполнена ⟨ЛЕВАЯ ЧАСТЬ 1⟩.

По определению ⨀⟨ЛЕВАЯ ЧАСТЬ 1⟩ равна бесконечному выражению:

(() ! (⟨ЛЕВАЯ ЧАСТЬ 1⟩ ! (⟨ЛЕВАЯ ЧАСТЬ 1⟩ ! (.....))).

Обычно в диалоговых системах молекулярной графики используется язык командного типа или меню. В данной системе применен более сложный язык как средство: упрощения и модификации молекулярных объектов; анализа сложных молекул, в частности белков. Рассмотрим пример:

NIF ⨀ ALL & B(S 2) & (TRP & ⨀ POL & TRP)
& E(S 2) : D (S 2) FW,

где ALL считывает из объекта S1 очередную аминокислоту в объект S10 и передвигает курсор. Седовательно, ⨀ ALL сопоставляется с произвольной цепочкой аминокислот;

(TRP & ⨀ POL & TRP) сопоставляется с произвольной цепочкой полярных аминокислот, ограниченной с обоих концов триптофанами (название аминокислоты);

B(S 2)&X& E(S 2) означает, что подцепочка, сопоставлявшаяся с образцом X, будет записана в объект S 2;

D(S2) — оператор вывода растрового изображения молекулярного объекта S2.

Таким образом, приведенный оператор осуществляет вывод изображений всех цепочек аминокислот, сопоставленных с образцом (TRP ⨀ POL TRP).

Возможности растрового вывода. Для повышения эффективности визуального изучения структуры молекулярных объектов система молекулярной графики должна обладать возможностями:

изображения модели произвольного типа, а также произвольной комбинации моделей;

окраски изображаемой поверхности в соответствии с интенсивностью поля;

полупрозрачного наложения двух произвольных изображений;

частичной модификации изображений;

усиления эффекта объемности моделированием второго источника света (эффект теней), уменьшением освещенности с глубиной, моделированием туманной дымки;

построения сечений объектов.

Гибкость при построении комбинированных и высокореалистичных изображений достигается введением Z-буфера — «полуфабриката» изображения, сохраняющего информацию о пространственной структуре изображаемого объекта.

Геометрические преобразования. Помимо обычных геометрических преобразований вращения вокруг заданной оси, перемещения и отражения система содержит набор преобразований типа «движение до выполнения некоторого условия». Такие преобразования используются при совмещении молекулярных структур для выяснения пространственного подобия (гомологий), сближения моделей молекул при моделировании химических взаимодействий (вы-

бор начального приближения для расчета энергетических характеристик).

Особенности реализации. Важное качество системы — расширяемость, которая обеспечивается созданием макроопределений и подключением к системе проблемных программ пользователя. Система реализована на языке Фортран на ЭВМ ЕС1061, соединенной с растровой системой PERICOLOR. Машинно-зависимые части выделены в отдельные небольшие по объему подпрограммы.

Телефон 3-38-19, Пущино

ЛИТЕРАТУРА

1. Johnson C. K. ORTEP: a Fortran thermal ellipsoid plot program for crystall structure illustration.— USA Oak Ridge National Laboratories, 1985.— 80 P. (Technical report N 3794).
2. Max N. L. Computer representation of molecular surfaces // IEEE Computer graphics and applications.— 1983.— August.— P. 21—29.
3. Нигматулин Р. С., Одеянко Б. Н. Комплексная машинная система графического вывода и анализа геометрии сложных молекулярных соединений // Автометрия.— 1978.— № 5.— С. 54—58.
4. Maugh T. Need a catalist? Design an enzyme // Science.— 1984.— V. 223.— P. 269—271.
5. Gund P., Adonese J. D., Rhodes J. B., Smith G. M. Threedimentional molecular modelling and drug design // Science.— 1980.— V. 208.— P. 1425—1426.
6. Боровина Т. А., Кислюк О. С. Интерактивная система получения изображений и анализа строения сложных молекул с помощью растровых графических устройств // Автометрия.— 1986.— № 5.— С. 114—116.

Статья поступила 22.03.88

УДК 681.3.06

В. П. Кротов

ЯЗЫКОВЫЕ СРЕДСТВА ДЛЯ ПРОГРАММИРОВАНИЯ РАСПРЕДЕЛЕННЫХ УПРАВЛЯЮЩИХ МИКРОСИСТЕМ

Специализированные микропроцессорные сети реального времени — интенсивно развиваемый ныне класс распределенных вычислительных систем. Особенности встроенных МП-систем сочетаются в них со специфическими чертами компьютерных сетей, что приводит к большим трудностям при разработке ПО. Примером могут служить системы управления телевизионными установками промышленного назначения [1]. Для разработки ПО таких систем необходима концептуально единая система программирования — разумная альтернатива традиционному набору средств и методов (ассемблеры, компиляторы, компоновщики, мониторы реального времени, многоуровневые модели сетей общего назначения).

Предлагаемый вариант — расширение языка Форт-83, хорошо зарекомендовавшего себя в области ПО управляющих микросистем [2, 3], легко дополняемого проблемно-ориентированными средствами. С точки зрения технологии разработки особенно ценны наличие в Форт-системах встроенного ассемблера и редактора, простота компиляции и малый объем. В расширение языка (Форт-Р) входят средства описания и создания программных процессов, обслуживание аппаратных прерываний, управление динамической памятью, установка логических соединений между удаленными процессами. Базовый механизм коммуникации процессов — буферизованный дуплексный канал с возможной селекцией сообщений. Поддерживаются также высокоуровневые протоколы обмена: рандеву языка Ада, запрос — ответ. Функционально-прикладные программы проектируются в виде совокупности взаимодействующих процессов, подходящим образом размещаемых в узлах сети. Отладка логики взаимодействия процессов может осуществляться на единственном инструментальном комплексе за счет унификации интерфейсов локальных и удаленных процессов. Обеспечена адаптация к потенциально широкому спектру целевых систем с различной топологией и аппаратурой связи.

Существующие реализации предназначены для разработки систем на базе МП К580 и К1801. Инструментальная среда Форт-Р-580: отладочные комплексы СО-04(05) либо другие, оснащенные ОС, совместимой с ISIS-II. Исполнительное ядро расширенной Форт-системы, устанавливаемое в каждый целевой узел, имеет объем 5,5...6 Кбайт. В изделиях достаточной сложности эти расходы компенсируются высокой компактностью прикладных программ.

Телефон 3-67-13, Новгород

ЛИТЕРАТУРА

1. Кротов В. П., Смирнов С. В., Холонмов А. Ю., Крутиков Н. И. Микропроцессорная система управления телевизионными установками // Микропроцессорные средства и системы.— 1987.— № 2.— С. 61.
2. Фаулджер Р. Программирование встроенных микропроцессоров.— М.: Мир, 1985.— 275 с.
3. Баранов С. Н., Ноздрунов Н. Р. Язык Форт и его реализации.— Л.: Машиностроение, 1988.— 157 с.

Сообщение поступило 21.03.88

УДК 681.32

В. В. Грушин, С. И. Запорожан, В. В. Новоселов, Д. В. Пузанков

ОДНОПЛАТНЫЙ МАТЕМАТИЧЕСКИЙ РАСШИРИТЕЛЬ ДЛЯ 8-РАЗРЯДНЫХ МИКРОЭВМ

С помощью одноплатного устройства с 32-разрядной внутренней архитектурой и таблично-микропрограммным вычислением функций резко повышается производительность микросистем на базе 8-разрядных микроЭВМ (при работе в реальном времени). По сравнению с программной реализацией математических зависимостей время вычислений сокращается в десятки (при передаче параметров из языка Паскаль) и сотни (при обращении на ассемблере) раз.

Среди 62 реализуемых функций — синус и косинус (время вычисления не более 80 мкс), арктангенс (84 мкс), экспонента (85 мкс), логарифм (74 мкс), корень квадратный (60 мкс), другие прямые и обратные тригонометрические, а также гиперболические функции над 32-разрядными операндами с плавающей точкой. Выполняется преобразование форм представления операндов для фиксированной и плавающей точки за 15 и 23 мкс. Обеспечен полный набор арифметических операций для каждой из форм представления чисел с временем выполнения от 1,6 до 27 мкс. Предусмотрены микропрограммные средства тестовой самодиагностики (2 мс).

Загрузка операндов из базовой микроЭВМ в математи-

ческий расширитель и чтение результатов осуществляются через 8-уровневый стек расширителя, реализованный на регистрах в составе микросхем типа КР1804ВС1. Наличие стека позволяет сократить число побайтовых пересылок промежуточных результатов прикладных вычислений между базовой микроЭВМ и расширителем. Верхушке стека соответствует адрес порта в пространстве ввода-вывода микроЭВМ; второй адрес закреплен за портом команд, третий за портом состояния расширителя. Обращения к портам выполняются в соответствии со стандартом шины И-41.

Совместимость с микроЭВМ СМ1800 обеспечивается форматами данных, интерфейсом и конструкцией (плата размером 233×220 мм), однако возможно применение устройства и в специализированных системах. Использованы микросхемы серий КР1804, К589, К556, К531, К555, К155 (ток потребления — не более 4 А от источника +5 В). Изделие снабжается подробным руководством пользователя и программными средствами проверки работоспособности в составе СМ1800.

197022, Ленинград, ул. Профессора Попова, д. 5, ЛЭТИ, кафедра вычислительной техники, тел. 234-25-03

Сообщение поступило 05.04.88

РЕКЛАМА

Оренбургский завод «Инвертор» изготавливает однофазные агрегаты бесперебойного питания АБП-24-230-2, 5-50-УХЛ4, гарантирующие безотказную работу микроЭВМ систем автоматического контроля, средств управления АСУ ТП, средств связи, радиорелейных сооружений, средств диспетчерского контроля, медицинской аппаратуры для жизнеобеспечения и диагностики.

Агрегат состоит из инвертора и выпрямителя, обеспечивает питание нагрузки от сети переменного тока напряжением 220 В. При падении напряжения до 190 В и повышении до 245 В питание нагрузки автоматически переключается с основного источника (сеть 220 В) на резервный (аккумуляторную батарею 24 В) с последующим преобразованием этого напряжения инвертором в переменное — напряжением 230 В.

Выпрямитель предназначен для заряда и подзаряда аккумуляторной батареи.

Технические характеристики

Полная выходная мощность, кВА	0,63
Номинальное входное напряжение от:	
основного источника, В	220
резервного источника, В	24
Номинальное выходное напряжение, В	230
Номинальный выходной ток, А	2,5
Время переключения нагрузки на резервный источник питания, с	0,04
Номинальная частота выходного напряжения, Гц	50/60
Габаритные размеры, мм	410×280×710
Масса, кг	53

Агрегаты могут поставляться как комплектно, так и в виде отдельных выпрямителей и инверторов.

Ввод агрегата в эксплуатацию, дальнейшее его обслуживание, а также обучение обслуживающего персонала предприятие берет на себя.

Наш адрес: 460858, Оренбург, ГСП, пр. Автоматики, 8, тел. 5-49-27

ПОЛОЖЕНИЕ

О ПРОВЕДЕНИИ ВСЕСОЮЗНОГО КОНКУРСА НА СОЗДАНИЕ ПЕРСПЕКТИВНЫХ МОДЕЛЕЙ БЫТОВЫХ ПЕРСОНАЛЬНЫХ ЭВМ

I. ОБЩИЕ ПОЛОЖЕНИЯ

1. Всесоюзный конкурс на создание перспективных моделей бытовых персональных ЭВМ проводится Государственным комитетом СССР по вычислительной технике и информатике (ГКВТИ, СССР), Центральным Комитетом ВЛКСМ (ЦК ВЛКСМ), Центральным Советом ВОИР (ЦС ВОИР), Центральным комитетом профсоюза рабочих радиоэлектронной промышленности (ЦК профсоюза) с 1 октября 1988 г. по 30 сентября 1989 г.

2. Целью проведения конкурса является создание перспективных высоконадежных и дешевых моделей бытовых персональных ЭВМ на отечественной серийно-выпускаемой элементной базе для их последующего массового производства.

II. УСЛОВИЯ КОНКУРСА

1. На конкурс принимаются действующие макеты персональных ЭВМ с базовым программным обеспечением.

2. В конкурсе могут принять участие коллективы объединений, промышленных предприятий, научно-исследовательских, конструкторских и других организаций, высших и средних специальных учебных заведений, студенческие конструкторские бюро, центры научно-технического творчества молодежи, первичные организации ВОИР, иные творческие коллективы, а также другие творческие формирования и самодеятельные авторы.

3. Представляемые на конкурс материалы должны быть выполнены в соответствии с требованиями действующих нормативно-технических документов.

4. Параметры действующих макетов должны отвечать ГОСТ 27201-87 «Машины вычислительные электронные персональные» (класс ПМ-1).

5. Материалы и действующие модели представляются под девизом до 31 августа 1989 г. в МНТК «Персональные ЭВМ» по адресу: 117900, Москва, В-334, ул. Вавилова, 30/6 с пометкой «На конкурс «Бытовые ПЭВМ» (без указания фамилий авторов).

Отдельно, под тем же девизом, с пометкой «Сведения об авторах» представляются материалы, содержащие наименование организации (адрес, ведомственная подчиненность, телефон), сведения об авторах-участниках конкурса: фамилия, имя, отчество, место работы, должность, специальность, домашний адрес, а также справки о долевом участии каждого члена авторского коллектива, заверенные в установленном порядке.

Материалы представляются только ценными почтовыми отправлениями. Датой поступления материалов на конкурс считается дата почтового штемпеля.

Подача материалов на конкурс не лишает авторов права представлять заявки в установленном порядке в Госкомизобретений при ГКНТ СССР для регистрации приоритета.

Ответственный секретарь оргкомитета несет ответственность за сохранность и анонимность представленных на конкурс моделей ПЭВМ и материалов к ним.

6. Для организации работ по проведению конкурса создается организационный комитет. Оргкомитет рассматривает разработки, представленные на конкурс, и выносит предложения о награждении авторов лучших работ. Для оценки представляемых на конкурс разработок организационный комитет имеет право привлекать для экспертизы специалистов различных объединений и организаций.

7. Для поощрения авторов и авторских коллективов — победителей конкурса учреждены премии ГКВТИ СССР, ЦК ВЛКСМ, ЦС ВОИР, ЦК профсоюза:

авторам (автору) макета бытовой персональной ЭВМ:
одна первая премия — профессиональная ПЭВМ ЕС1841;
две вторые премии — профессиональная ПЭВМ «Электроника-85»;
три третьи премии — туристические путевки по линии Бюро молодежного международного туризма «Спутник».

8. Государственный комитет СССР по вычислительной технике и информатике, Секретариат ЦК ВЛКСМ, Президиум ЦС ВОИР, Президиум ЦК профсоюзов обеспечивают все заинтересованные организации своевременной информацией об условиях проведения конкурса.

Организации-консультанты:

Институт проблем информатики Академии наук СССР (117900, Москва, В-334, ул. Вавилова, 30/6, ответственные — Христачевский С. А., Гуткин М. И., телефон для справок 238-40-66);

Центр компьютерной техники «Юность» (129223, Москва, ВДНХ СССР, павильон № 3, ответственный — Фомин В. В., телефон для справок 187-99-95.

9. Координация действий по организации конкурса, соблюдение условий конкурса и работы оргкомитета возлагаются на ответственного секретаря оргкомитета.

III. ПОДВЕДЕНИЕ ИТОГОВ КОНКУРСА

1. Итоги конкурса подводятся решением оргкомитета, утвержденного постановлением Государственного комитета СССР по вычислительной технике и информатике, ЦК ВЛКСМ, ЦС ВОИР, ЦК профсоюза, до 30 сентября 1989 г.

2. При подведении итогов конкурса оргкомитет учитывает: технологичность разработки; ее соответствие ГОСТ; возможность использования имеющегося в стране программного обеспечения; стоимость ПЭВМ; отсутствие остродефицитных изделий электронной техники.

3. По каждой ПЭВМ оргкомитет составляет письменное заключение, дает рекомендации о дальнейшей реализации разработок, занявших призовые места.

IV. ОРГАНИЗАЦИЯ ОСВОЕНИЯ СЕРИЙНОГО ПРОИЗВОДСТВА БЫТОВЫХ ПЕРСОНАЛЬНЫХ ЭВМ, ПРЕДСТАВЛЕННЫХ НА ВСЕСОЮЗНЫЙ КОНКУРС

1. Государственный комитет СССР по вычислительной технике и информатике, Межотраслевой научно-технический комплекс «Персональные ЭВМ», Министерство радиопромышленности СССР, Министерство электронной промышленности СССР, Министерство приборостроения, средств автоматизации и систем управления СССР и другие заинтересованные министерства и ведомства обеспечивают быстрейшее внедрение в производство бытовых персональных ЭВМ, занявших в конкурсе призовые места.

2. Представленные на конкурс макетные образцы и соответствующие материалы, не занявшие на конкурсе призовых мест, возвращаются авторам в месячный срок после завершения конкурса.

3. Модели бытовых персональных ЭВМ и комплект документации, отмеченные премиями конкурса, а также наиболее интересные решения других разработок с согласия авторов направляются в заинтересованные министерства и ведомства для разработки рабочей документации и освоения в серийном производстве.

4. Лучшие разработки, отмеченные оргкомитетом конкурса, будут рекомендованы для экспонирования на ВДНХ СССР в одном из разделов постоянно действующей выставки «Самодеятельное техническое творчество».

УДК 681.3

С. М. Иванов, В. А. Шахнов

ДОСТИЖЕНИЯ НРБ В ОБЛАСТИ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Недавно в Москве под эгидой ассоциации «Электроника» (НРБ) и секретариата СЭВ состоялась специализированная выставка «Микрокомпьютерная техника и информационные сети». Организаторами выставки стали известные в НРБ и во многих странах мира болгарские хозяйственные объединения «Программные продукты и системы» (ХО «ППС») г. София и «Микропроцессорные системы» (ХО «МПС») г. Правец. Выставка способствовала установлению прямых связей между заинтересованными организациями, определению направлений развития, исходя из интересов потребителей, подписанию хозяйственных договоров и контрактов и др.

Объединение «Программные продукты и системы» состоит из 22 связанных между собой единой структурой предприятий, решающих проблемы информатизации социальной и экономической жизни НРБ.

Центральный институт программных продуктов и систем в Софии проводит исследования в области информатики и системного инженеринга, анализа маркетинга, разрабатывает базовые и прикладные продукты, различные программно-аппаратные интерфейсы, периферийные и коммуникационные контроллеры, осваивает и разрабатывает технологические средства программирования.

В объединение входят четыре технологических комбината:

«Национальный программный и проектный фонд» в Софии обеспечивает подготовку и издание документации, распространение, установку на месте и сопровождение программных продуктов и систем. Комбинат регистрирует создаваемые в стране программные и программно-аппаратные продукты, предоставляет информацию о них и консультирует пользователей. Осуществляет комплексные поставки программных и аппаратных средств для разных видов компьютерных технологий.

Комбинат «Прикладные системы» в Софии производит многослойные (до 16 слоев) печатные платы с толщиной шин до 150 мкм, монтирует электронные компоненты, осуществляет сборку и наладку средств вычислительной техники, выпускает специализированные прикладные системы в области коммуникаций, проектирование, обработку и изображений.

Комбинат «Комплектовка и программно-

технические услуги» в Софии комплектует программные средства с аппаратными, поставляемыми болгарскими и зарубежными фирмами и предлагает системы «под ключ». Выполняет установку систем, осуществляет гарантийную и послегарантийную поддержку. Разрабатывает и поставляет системы для автоматизации проектирования, внутризаводского транспорта, управления обрабатывающими машинами — основные составляющие компьютерно-интегрированных производств.

«Национальный учебный центр по информатике» в Софии проводит по заказам курсы обучения персонала пользователя.

Семнадцать предприятий-производителей программных продуктов вносят свой вклад в осуществление узких контактов объединения «ППС» с пользователями, распространяют продукт всего объединения, поставляют программно-технические комплексы, осуществляют гарантийную и послегарантийную поддержку. Наряду с этим каждое предприятие имеет свою специфику, расширяющую круг прикладных областей, в которых используется продукция объединения.

Направление «Информатика на предприятии» охватывает программные продукты, связанные с проектно-конструкторской деятельностью, планированием, технологической подготовкой и оперативным управлением производством, процессами изготовления продукции, сбытом и реализацией продукции, материально-техническим снабжением, складским хозяйством, финансово-бухгалтерской деятельностью, трудом и заработной платой, управлением кадрами, делопроизводством.

Для проектирования изделий предлагаются универсальные графические системы конструирования и автоматизированного документирования: СИСКО, УНИКАД, АОС-16 и др. Их использование в машиностроении и электронике значительно облегчается наличием созданных заранее библиотек основных графических символов, а также других технологических средств для проектировщика, например, КОНСИС и ТЕХНОЛОГ.

Имеется возможность ввода в компьютер изображений с помощью цветной или черно-белой видеокамеры, для ввода и обработки графической информации.

В предлагаемых системах процесс проектирования часто связан с процессом производства. Пример — автоматизированная система подготовки программ для газовой, плазменной и лазерной резки ПРОМЕТЕЙ. С ее помощью можно добиться ускорения процесса технической подготовки раскроя листового материала, сокращения материальных и трудовых затрат и повышения качества готовой продукции.

Базовая система ТЕХНОКАД с помощью связанных с ней продуктов ТЕХНО-NC, ТЕХНО-РСВ, ТЕХНО-МКЕ и ТЕХНО-СПЕЦ позволяет автоматизировать управление металлорежущими станками с числовым программным управлением, проектирование печатных плат и изготовлению спецификаций.

Значительное число программных продуктов и систем предназначено для организационного управления производством. Это интегрированная информационная система для управления небольшими и средними предприятиями с помощью сети персональных компьютеров СИНХРОН-16, диалоговая система ПЛАН-ОТЧЕТ 16, ТПП — для технологической подготовки производства и др.

Программные продукты МИКРОСКЛАД, МТС, ВСС реализуют комплексное решение управлением складским хозяйством. Применяется для управления высокостеллажными системами и стеллажными блоками с произвольными характеристиками.

Система СТРАИД-ТОТАЛ включает семь программных продуктов, которые автоматизируют деятельность по учету заработной платы, основных средств, материалов, поставщиков.

Ряд программных продуктов находит применение в областях проектирования строительства, здравоохранения, транспорта, издательской деятельности, курсов обучения.

Программный комплекс ОПТИМУМ включает программные системы для экономико-математического моделирования. Одна из них КСИ ОПТИМА — для оптимизации с помощью различных современных математических моделей и методов: LP-16 — линейное, ДР-16 — дискретное, FP-16 — потоковое, СР-16 — комбинаторное программирование.

Разработаны технологические программные средства, создающие условия для развития программной индустрии, обеспечивающие технологическую базу, необходимую для создания новых программных продуктов и приложений.

N-BASE и B-BASE дают возможность создавать информационные базы различных типов; с помощью ИНТЕРЭКСПЕРТ можно разрабатывать прикладные системы с базами данных и базами знаний; ОКНО и МАГИУМ — технологические средства при программировании на языке «С»; с помощью ИНФОС создаются, поддерживаются и используются информационные базы, содержащие большие объемы документальной и текстовой информации. Программная среда для персональных компьютеров МУЛТИПРО обеспечивает эффективные средства разработки прикладных программных систем.

Усилия болгарских специалистов направле-

ны на создание таких программных продуктов, которые можно интегрировать в иерархические системы. Существенная роль в этом направлении принадлежит коммуникациям между микроЭВМ по коммутируемой телефонной сети, локальным сетям.

Объединение «Микропроцессорные системы» (г. Правец) располагает разработками в области персональных компьютеров и созданных на их базе микропроцессорных систем. Основой научно-технической политики Объединения «Микропроцессорные системы» является гамма 8-, 16- и 32-разрядных персональных компьютеров и модулей для них, системы для автоматизации конструкторской и конторской деятельности на базе 8- и 16-разрядных микрокомпьютеров и микропроцессорные модули и системы для автоматизации промышленных процессов. Среди разработок объединения «Микропроцессорные системы» 16-разрядный персональный компьютер ПРАВЕЦ-16 (ЕС 1839) с оперативной памятью 640 Кбайт, ЗУ ГМД, ЗУ МД с емкостью 10/20 Мбайт; высокопроизводительный 16-разрядный персональный компьютер ЕС 1838 с оперативной памятью виртуальной организации, емкостью до 3 Мбайт, ЗУ ГМД емкостью 360 Кбайт/1,2 Мбайт, ЗУ МД емкостью 10/20/30 Мбайт.

Их совместимость с персональными компьютерами APPLE и IBM богатое программное обеспечение дают возможность создавать микропроцессорные системы и информационные сети с различными сферами применения.

Микрокомпьютер ПРАВЕЦ-8Д — первый в НРБ персональный компьютер бытового назначения. Отличные технические характеристики дают возможность широко применять его в области образования, научных исследований и инженерных работ. Для усовершенствования компьютера разработана дисковая операционная система, построенная на базе ПРАВЕЦ-8Д и накопителя на ГМД, увеличивающего внешнюю память компьютера и позволяющего работать с операционной системой MDOS.

В образовательных целях Объединение «Микропроцессорные системы» предлагает систему обучения «Компьютерный класс», оснащенную соответствующим комплексом аппаратных средств и программного обеспечения, позволяющим проведение современного обучения, самообучения и проверки знаний. Система работает под управлением операционной систем ПДОС-16, которая осуществляет связь восьми компьютеров типа ПРАВЕЦ-82 или ПРАВЕЦ-16 с главным компьютером типа ЕС 1839 или ЕС 1838.

Для автоматизации управленческой деятельности, производства, научных исследований и разработок большое распространение во всех отраслях народного хозяйства получили локальные сети микрокомпьютеров.

МИКРОЛИМ — локальная сеть с шинной топологией, которая построена на базе персональных компьютеров типа ЕС 1938 (ПРАВЕЦ-286), ЕС 1839 (ПРАВЕЦ-16А) и IBM PC/XT/AT и предназначена для автоматизации административно-управленческой деятельности. Скорость передачи по линии связи 10 Мбайт/с. Для файлового обслуживания в сети разработан специализированный сетевой процессор. Номинальное число рабочих станций, работающих под управлением одного сетевого процессора, равно восьми.

МИКРОРИНГ — низкоскоростная гетерогенная локальная сеть с кольцевой топологией, построенная с помощью последовательно соединенных коммуникационных адаптеров. Максимальная длина круга — 6 км. Скорость передачи данных по кругу 125/250 Кбайт/с. В состав локальной сети входят персональные и профессиональные компьютеры типа IBM PC/XT, ПРАВЕЦ-8М, мини-ЭВМ и различные периферийные устройства. Используется для построения распределенных вычислительных систем, систем административного обслуживания, учебных комплексов и др.

Коммуникационная система МИКРОСТАР, созданная на базе локальной сети с топологией типа «звезда». С помощью одного центрального узла обслуживаются восемь периферийных. Скорость обмена данными между узлами зависит от расстояния между ними: 115200 бит/с на расстоянии 15 см и 19200 бит/с на расстоянии более 300 м.

В локальной сети МИКРОЛИМ существует возможность радиосвязи между сервером одного сегмента и рабочей станцией другого сегмента. Таким образом, осуществляется связь между двумя сетями. В локальной сети МИКРОРИНГ два узла могут быть соединены линией оптической связи, что обеспечивает новые возможности при применении сети в промышленности и для специальных целей.

Система автоматизации административно-управленческой деятельности и проектирования построена на базе локальной сети МИКРОЛИМ, связанной с ЕС ЭВМ. Разработана программная система специализированного терминального процессора для связи локальной сети с ЭВМ.

Локальная сеть U-LAN обеспечивает обмен данными между пользователями сети, передачу по сети данных, содержащих частную или служебную корреспонденцию (электронная почта). Конфигурация системы — до 63 компьютеров ПРАВЕЦ-8, отсутствие управляющего компьютера, длина кабеля до 500 м.

Локальная сеть ISNET предназначена для подключения персональных компьютеров се-

мейства ПРАВЕЦ-16, IBM PC/AT и XT; разработана в двух вариантах. Максимальное число подключенных компьютеров — 255 при длине кабеля 750 м. Обслуживается из центральной станции, обрабатывающей ее ресурсы. Находит применение в административно-управленческой деятельности и при быстром обмене данных.

В области специализированных рабочих мест для автоматизации инженерного труда в машиностроении и электронике была представлена графическая система, созданная на базе 16-разрядных персональных компьютеров типа ПРАВЕЦ. Система имеет высокую производительность, большой объем оперативной и внешней памяти и видеоконтроллер, совместимый с EGA с максимальной разрешающей способностью 800×600 точек или видеоконтроллер для передачи графических изображений с высокой разрешающей способностью, дает возможность ввода, обработки, сохранения и вывода графической информации в системах автоматизации учрежденческой, инженерной и научной деятельности, работает с матричными 9-игольными печатающими устройствами, графопостроителем, диджитайзером и преобразователем координат типа мышь.

Автоматизированная система управления торгово-бухгалтерскими процессами и складским хозяйством в универмаге разработана на базе ПЭВМ ПРАВЕЦ-16. Работает с кассовыми автоматами типа ЭЛКА-90. Организована в четыре круга, каждый из которых управляет 16 электронными кассами. Система широко применяется в торговой сети.

Рабочая станция с искусственным интеллектом, основанная на персональном компьютере ЕС 1839, является относительно дешевым и доступным средством. Она обеспечивает оператору среду для использования, базирующуюся на оригинально разработанной версии языка ПРОЛОГ, экспертную систему управления процессами КОРТЕКС, симуляционный пакет на основе обобщенных сетей, консультирующую систему РИСК, ориентированную на торговых служащих, и экспертную систему для сельского хозяйства АГРОТЕКСТ. Эта среда может легко настраиваться пользователем.

Большой интерес вызывает интегрированная система управления внедренческим звеном, обеспечивающая автоматизированное управление подразделениями и институтами в условиях работы на хозрасчете. В систему включены следующие подсистемы, для которых разработаны прикладные программы:

СКЛАД — оперативный учет и ведение складского хозяйства на предприятиях;

ЭЛЕКТРОН — автоматизация процессов, связанных с изготовлением документов, необходимых для планирования и заявок импортных, электронных элементов;

ПЛАН — формирование и контроль плана научно-исследовательской и технологической деятельности и его отчета и другие информационные системы для автоматизации наблюдения за основными средствами наблюдения, документирования, оплаты дополнительных вознаграждений и изготовления необходимых документов, автоматизации деятельности отдела кадров, формирования и наблюдения за штатом предприятия.

Центр кибернетических технологий и систем в области сельского хозяйства и консервной промышленности АГРОАСУ располагает следующими системами:

— организационно-экономического управления сельскохозяйственным предприятием — целостная компьютерная система для автоматизации функций нормативной базы, планирования, отчета и анализа и оперативного управления одним сельскохозяйственным предприятием для удовлетворения потребностей всех уровней системы управления;

— оптимизации производства в сельском хозяйстве, которая охватывает и регистрирует знания высококвалифицированных специалистов в области прогнозирования производства в растениеводстве и делает их доступными для широкого круга сельскохозяйственных рабочих в виде советов для решения конкретных проблем, возникающих в практических ситуациях. На базе экспертной оценки система прогнозирует объем производства, обеспечивая возможность стимулирования ситуаций факторами, оказывающими влияние на формирование производства.

АГРОАСУ и ВНИПТИК (Москва) разрабатывают совместно «Автоматизированную систему информационно-вычислительного обслуживания» (АСИВО) в Шпаковском районе Ставропольского края. АСИВО охватывает деятельность по автоматизации процессов управления сельскохозяйственными предприятиями и научного обслуживания производства.

Продукция болгарских объединений отвечает всем требованиям Единой системы электронно-вычислительной техники социалистических стран — ЕС ЭВМ и Системы мини-ЭВМ — СМ ЭВМ. Это является основой плодотворного сотрудничества в рамках СЭВ в области компьютерной техники и информационных систем.

Телефон 252-52-11, Москва, Шахнов В. А.

Статья поступила 27.02.89

УДК 681.3.06-181.4

Е. В. Лишак

БАЙТОТОРГОВЛЯ: ПРОШЛОЕ, НАСТОЯЩЕЕ, БУДУЩЕЕ

Чтобы понять, какова сейчас практическая, я бы сказал, экспериментальная экономика информатики, ознакомимся с двумя цитатами: **«Налет токийской полиции на фирму-изготовителя программного обеспечения.** Полиция Токио совершила налет на восемь контор фирмы Ma's Marketing с целью проверить утверждение о том, что эта компания скопировала и продала 3000 видов программ для персональных компьютеров, права на которые принадлежат фирме IBM и нескольким крупным программотехническим фирмам Японии. Компания Ma's Marketing обвиняется в продаже скопированных программ, предназначенных для персональных компьютеров PC фирмы IBM, по ценам примерно в пять раз ниже цен, установленных фирмой IBM [1]». **«Тяжба между двумя японскими фирмами по вопросу об авторском праве.** Представители компании NEC Corp. (Токио) утверждают, что компания Seiko Epson (Сува), выпустившая недавно в продажу новый персональный компьютер PC-286, совместимый с пользующимся в Японии наивысшим спросом изделием PS-9801 компании NEC, нарушила ее право собственности на программные средства. В этом заключена немалая доля иронии, поскольку сама компания NEC является в настоящее время ответчиком по аналогичному судебному иску в США... [2]». Как видно, экономика информатики уже выходит из коротких штанишек натурального хозяйства. Программный продукт стали делать не только для себя, но и для других. А как продавать?

Приведенные цитаты говорят о том, что продавать информацию — дело трудное. На первый взгляд, естественное, но, видимо, неверное решение — эту трудность преодолеть силой, т. е. совершенствовать правовую охрану собственности на информацию [3]. Ниже я попытаюсь показать, что такой путь обречен на неудачу (да простят мне это авторы [3], тем более, что я и сам не знаю, как выйти из создавшегося положения).

Написанное ниже не содержит конструктивных предложений по «построению экономики информатики». Я только попытался выделить некоторые закономерности, позволяющие эту проблему проанализировать с одной из возможных сторон. Большую помощь в этом мне оказал Л. Л. Петрович. Он дал мне начальное (послевузовское) экономическое образование, но не несет ответственности за все нижеизложенное.

«Служил Гаврила программистом». Некоторые говорят, что теперь программисты свои программы не пишут, а производят. Ну что ж, давайте проанализируем, чем отличается изготовление программы для ЭВМ от, скажем, испекания булки.

Рассмотрим процесс производства продукции (неважно какой: программного продукта, видеофильмов, столовых ложек, автомобилей или телефонных справочников) с информационной точки зрения. Этот процесс можно разбить на две стадии: *разработка* (итог этой стадии — комплект документации, опытный образец, модель, эталон и т. п.) и *изготовление* (получение некоторого числа экземпляров изделия).

При *разработке* накапливается информация о будущем продукте и создается носитель (или несколько разных носителей) с информацией, необходимой для изготовления продукта. *Примеры* носителей — штамп столовой ложки, документация на изготовление хлебобулочного изделия, типографские матрицы телефонного справочника, технологическая и конструкторская документация, дискета с программами для станков с ЧПУ, магнитная лента с подлинником программного продукта, оригинал видеофильма.

При изготовлении эта информация (или только часть ее) переносится на материальные носители, которые вместе с этой информацией и являются *изделиями*. С информационной точки зрения стадия изготовления — это копирование.

Итак, производство ПО и булки ничем не отличается? Пусть так.

Однако экономика — это свойства не только производства, но и распределения. А вот эти-то свойства у булки и программы ЭВМ существенно разные. В этом смысле, если программу и можно сравнивать с изделием хлебопекарной промышленности, то только с теми пятью хлебами, которыми Христос накормил всех. Программы и булки распределяются в обществе по разным законам, и законы эти определяются еще на стадии производства, что я и попытаюсь показать ниже.

С чем «потребляют» информацию? Съедая булку, «потребляем» ли мы вместе с ней ее рецепт? Если не очень голодны, и у нас есть выбор, то мы будем есть не любую булку. Мы захотим попробовать новый сорт булок. Из чистого любопытства. И, съев булку-незнакомку, мы, можно сказать, «потребили» информацию, содержащуюся в рецепте. Но не всю.

Рецепт изготовления булки, полученный на стадии разработки изделия, состоит из двух принципиально различных (для данного потре-

бителя) частей. Вместе с булкой мы получили только ту часть информации, содержащейся в рецепте, которую в состоянии воспроизвести, выпекая подобную булку к приходу гостей. Назовем эту часть потребительной информаций. С тех пор, как изобретатель булки с изюмом продал свой первый экземпляр, он не является более собственником этой идеи (если клиенты знают, что такое изюм).

Другая часть информации осталась у хлебопека. Мы съели булку, поняли, что вкус ее необычен, приятен и т. п. Причина этого осталась нам неизвестной. Значит, эту часть рецепта нам «съесть» не удалось. Назовем такую информацию технологической. Отсутствие технологической информации лишает изделие свойства копируемости.

Пример — племенной бык. Потребительная информация — генетический код.

Еще пример — кубик Рубика. Для потребителей, знакомых с ним, потребительной информации в кубике нет. Для тех, кто не знаком с принципом его устройства, структура кубика информативна.

Последний пример иллюстрирует очень важное обстоятельство. Пока изделие представляет собой новинку, информация, полученная на стадии разработки, копируется (на стадии изготовления). Для традиционных же изделий изготовление — это копирование данных о структуре изделия, не содержащих полезной (потребительной) информации.

Этот пример интересен и тем, что для потребителя, технологически не подготовленного к пониманию устройства кубика, устройство его останется тайной. Для такого потребителя заключенная в этом кубике информация будет не потребительной, а технологической (взаимоотношения Пятницы с ружьем Робинзона).

Три угла экономики знаков. Будем называть знаком современный продукт производства, несущий в себе информацию и в то же время содержащий в себе материальный ее носитель [3]. В процессе производства знаков затраты делаются, как мы уже выяснили, по трем направлениям: изготовление материального предмета (копирование информации на носитель); разработка технологии; разработка потребительной информации.

Каждый из трех видов затрат определяет соответствующую компоненту стоимости конечного изделия. Реальное изделие имеет все три стоимостные компоненты, поэтому мы будем рассматривать его в виде комплекса из трех гипотетических изделий: простой товар, технология, символ.

Простым товаром (короче — **п-товаром**) назовем изделие, обладающее следующими свойствами: процесс производства состоит только из этапа изготовления (копирования), т. е. этапа разработки для п-товара не существует.

Простой товар — результат простого труда (в терминологии [4]), т. е. труда, не требующего никаких навыков, знаний, технологии. Строго говоря, такого п-товара быть не может, так как товарное производство возникло из разделения труда, предполагающего определенные навыки (т. е. наличие технологической информации) у производителя. Но мы, ведь, договорились, что п-товар — это некоторое допущение (аналогичная абстракция — простой труд — с успехом используется в классической политэкономии).

Технология — это весь набор навыков, технологических секретов, профессионального чутья, таланта, интуиции, который отличает сложный труд от простого (в терминологии [4]). Гипотетическое изделие, стоимость которого заключена лишь в технологии, мы назовем авторизованным товаром (или а-товаром). Пример хорошего приближения к а-товару — произведение живописи.

Итак, экономика информатики началась, когда появилось общественное разделение труда (тогда вообще возникла товарная экономика). Первоначально технологическая информация накапливалась в человеческом сознании, теперь же она отчуждается от человека и переносится им на более дешевые и удобные носители [5].

Очевидно, что обычный товар [4] — это «смесь» из п-товара и а-товара. Современные изделия в массе своей, увы, еще более сложны благодаря опять же информации, но уже потребительной.

Символом для краткости мы будем называть потребительную информацию, а точнее, гипотетическое изделие, стоимость которого заключена лишь в разработке потребительной информации. Пример хорошего приближения к символу — магнитная лента с программным продуктом, если стоимость продукта значительно превосходит стоимость носителя. Очевидно, что символ обладает стоимостью. Ведь при его разработке использовался труд и средства производства. Беда лишь в том, что символ невозможно продать, но об этом ниже.

Знак. Итак, мы ввели некоторые абстракции. Это п-товар, нечто сугубо материальное, несущее в себе труд по преобразованию материи, но абсолютно свободное от информации; а-товар, обязанный своим происхождением технологической информации, не предназначенной для потребителя. И, наконец, символ — информация, без труда потребляемая, но создаваемая в поте лица.

Реальные же, знакомые нам всем изделия назовем знаками. Они несут в себе свойства рассмотренных выше абстракций.

Примеры. 1. Книга в допечатные времена ближе к товару. Изготовление носителя и копирование требуют иногда даже больших затрат, чем создание потребительной информации (сочинение).

2. Книга в условиях свободного доступа к копировальной технике, дешевизны бумаги и наличия устройств оптического ввода данных в ЭВМ более похожа на символ.

3. Программный продукт с документацией на машинном носителе еще ближе к символу.

4. Племенной бык. Такой знак имеет существенные черты и символа и товара. Репродуцирование быка с почти таким же геномом — очень дорогое мероприятие, хотя и более дешевое, чем получение («разработка») первого экземпляра.

Интересно, что и в животноводстве экономическая ситуация стала чуть ближе к той, что имеет место в программировании. Уже стало классикой вместо такого носителя информации, как бык, использовать более дешевый носитель — его семя. А в последнее время вопрос решен и с коровами. Потребителю поставляются эмбрионы для трансплантации их обычным (неплеменным) коровам [6]. «Копирование быка» упростилось. Но когда генная инженерия даст возможность копировать гены вообще без участия животных, селекционеры окажутся в той же ситуации, в которую сейчас уже довольно часто попадает, скажем, фирма IBM.

5. Программный продукт, требующий сопровождения разработчиком,— во многом а-товар.

6. Модные джинсы — более символ, чем товар, поскольку несут информацию (потребительную) о превосходстве хозяина брюк над окружающими, и эта информация может обойтись потребителю гораздо дороже, чем ее материальный носитель (сами брюки).

7. Магнитофонная запись — это, в основном, знак, со свойствами символа. Внедрение цифровых систем записи и воспроизведения на магнитной ленте лишает носитель с такой записью свойств товара, к большому огорчению фирм звукозаписи, спасает которые пока только отличная от нуля стоимость носителя и цифровых устройств воспроизведения. Следующая цитата послужит иллюстрацией к сказанному. **«Борьба вокруг цифровых звуковых кассет.** Яблоком раздора между крупнейшими японскими изготовителями бытовой аппаратуры и международными звукозаписывающими компаниями стало соблюдение авторских прав на копирование. Звукозаписывающие компании настаивают на введение в устройства ... специальных ИС, которые бы предотвращали копирование материалов, имеющих особые коды. Однако в декабре 1986 г. руководство Японской ассоциации электронной промышленности отвергло это предложение. Аналогичный законопроект был отклонен конгрессом США в 1986 г., как и закон, устанавливающий 35 %-ный тариф на ДАТ-устройства, неоснащенные защитными ИС [7]».

8. Экспертная система или база знаний на машинном носителе очень близка к символу.

9. Теорема Пифагора может считаться идеальным символом. На базе знания этой теоремы построена геодезия и топография. Несомненно, теорема приносит огромный экономический эффект. Но никто из тех, кто по длине катетов с большой пользой для себя находит длину гипотенузы, эту теорему не покупал.

10. Произведение искусства — это, скорее, а-товар. Можно считать мастерство и талант художника технологической информацией.

Но музыкальная мелодия для такого класса потребителей, как эстрадные певцы или вокально-инструментальные ансамбли, более похожа на символ. Один раз услышанная ими музыкальная тема может уже служить им как средство производства.

Товарообмен, известный всем нам,— это устойчивое общественное отношение. В процессе обмена труд, затраченный на изготовление обмениваемого экземпляра, полностью переходит от производителя к потребителю в обмен на эквивалентный труд. Государство не формирует при помощи права отношения товарообмена, а лишь облегчает последние, обеспечивая законом свободу товарообмена. Отметим свойства товаров, которые затем сравним с соответствующими свойствами символов. Во-первых, производство товара индивидуально (индивидуальны затраты на производство каждого товарного изделия). Во-вторых, потребление товарного изделия также индивидуально (в условиях государственной собственности на средства производства положение не меняется, так как потребление средств производства — это прежде всего использование их. А использует, скажем, некоторый прокатный стан не государство, а конкретный коллектив работников предприятия).

Знакообменом будем называть обмен символа на товар, так как другие виды знакообмена тривиальны. Сопоставим черты символа с соответствующими чертами товара: во-первых, производство изделия-символа, как и товарного изделия, индивидуально; во-вторых, потребление изделия-символа, в отличие от товарного изделия, общественно.

Одну и ту же песню поют и слушают много «потребителей». Одной и той же теоремой Пифагора пользуются многие. Компенсировать затраты производителю, таким образом, должно все общество. Но ни один из потребителей символа, имея полную возможность символ

этот потреблять, не находится в особом положении, которое экономически (без вмешательства государства и закона) вынуждает его компенсировать производителю затраты на разработку.

Налицо противоречие между характером производства и характером потребления символа. Характер потребления в рамках существующих экономических формаций не обеспечивает товарной компенсации производителю символа столь же автоматически, как это происходит с производителем товара. Таким образом, это противоречие — антагонистическое и должно привести к смене формации. Товарные отношения и товарное производство стали препятствием на пути дальнейшего развития общества. Назовем это противоречие основным противоречием знаковой экономики (знакообмена). Оно делает обмен символа на товар неестественным, а стоимость, как меру на множестве символов, неопределенной (стоимость разработки символа по-прежнему существует, но стоимость каждого его экземпляра не имеет смысла).

Право при знакообмене идет вразрез с естественными побуждениями потребителей. Закон не защищает свободу обмена, а, наоборот, навязывая ограничения (авторское и патентное право), эту свободу отнимает. Применение его ограничено, и хорошо лишь на стадии развития информационного производства, когда объем производства символов существенно меньше объема производства товаров. Попытки применения права для сглаживания основного противоречия знакообмена неперспективны, так как всем членам общества вскоре придется стать либо прокурорами, либо адвокатами.

Так как мы не имеем возможности ждать, пока это противоречие приведет к смене экономической формации, то нам надо пытаться эту формацию прогнозировать и изобретать способы эффективного сглаживания этого противоречия в условиях имеющейся экономической формации. Займемся этой последней проблемой.

Анализ существующих способов знакообмена. Рассмотрим основные способы сглаживания отмеченного нами противоречия.

«Товарная нагрузка» или «племенной бык». Символ снабжается (возможно, искусственно) дорогим носителем, придающим знаку существенные свойства товара. Пример — ПО, «защитое» в интегральную схему. Отметим, что для общества искусственное удорожание носителя столь же накладно, как и увеличение числа прокуроров, в то время как естественная дороговизна носителей и копирования — это пока спасение производителей символов.

«Реклама». Символ с положительным товарным эквивалентом (фильм) соединяется с символом, имеющим отрицательный (!) товарный эквивалент (реклама). Все вместе имеет нулевой товарный эквивалент и распространяется, как и положено символу, по потребности.

«Автодорога» ... Используется свойство символа повышать потребительную стоимость товара, совместно с которым этот символ применяется. Например, большой выбор дорог и ПО повышает потребительную стоимость соответственно автомобилей и персональных ЭВМ (с этим ПО).

Возникает вопрос, можем ли мы, производя ЭВМ, финансировать разработчика ПО за счет того, что ЭВМ, укомплектованные этим ПО, оказались очень полезными? Увы, не можем, потому что из повышения потребительной стоимости ЭВМ совсем не следует более высокая цена этих ЭВМ.

Цена товара определяется не тем, как «сильно» он нужен, а тем, во что обходится обществу его производство. Если мы поднимем цену на нашу ЭВМ (чтобы было чем заплатить разработчику программ), то будут покупать не нашу ЭВМ, а более дешевую ЭВМ, сделанную другим производителем, причем, нарочно сделанную программно совместимой с нашей ЭВМ.

Аналогия из области товарообмена — система «автомобиль—дорога» (если, конечно, дороги бесплатны, как в нашей стране).

«Мода», «Новинка» или «Сенсация». Первые экземпляры символа обмениваются на товар, так как у потребителя может возникнуть целесообразность иметь символ как можно скорее (раньше других). Метод работает, если время (стоимость) копирования символа существенно отлично от нуля. Например, пока наша легкая промышленность осваивала импортные технологии для выпуска изделий из болоньи и кримплена, спрос на них прекратился.

Здесь, пожалуй, заключена самая сильная позиция товарной экономики в борьбе за свою жизнь. Дело в том, что символ становится достоянием всего общества не сразу, а по истечении некоторого времени. Если в модель отношений знакообмена включить параметр время, то исследование таких моделей «экономики переходных процессов» даст возможность частично перенести меру стоимости с множества товаров на множество знаков.

Переход от анализа стационарных рыночных процессов (закон стоимости) к анализу переходных рыночных процессов позволит выделить некоторый класс знаков, затраты (на разработку, копирование) и экономическая эффективность которых связаны определенными соотношениями. На класс этих знаков можно будет перенести меру стоимости. Но этот класс будет все более и более сужаться из-за «ка-

тастрофической» для него скорости удешевления копирования.

«Дар» — это основной способ распространения качественного ПО в нашей стране. Так происходит из-за специфики стимулирования труда и социального обеспечения программистов — программист, пишущий хорошие программы, все равно не может существенно изменить свое материальное положение (включая сюда, скажем, и положение в очереди на квартиру или телефон).

«Жрец» или «Сопровождение». Используются а-товарные свойства знака либо знак искусственно снабжается такими свойствами.

Приведенные выше способы сглаживания основного противоречия основаны на экономических методах. Ниже приведем несколько способов, связанных с правом, понимая под этим не только законы, но и прочие действующие в обществе нормы и правила, возмещающие производителю его затраты на производство символов. Вопросы защиты неимущественных прав производителя здесь рассматривать не будем.

«Патент». Символ объявляется собственностью (автора, другого лица, государства), что придает символу свойства а-товара. Но поддержание этих свойств требует от общества все более существенных расходов на юридическую деятельность. С появлением таких «почти идеальных» символов, как программный продукт и цифровая звукозапись, эти расходы стали приближаться к своему пределу, не приводя к достижению поставленных целей.

«Гонорар». Закон определяет автора и размер гонорара. Однако этот закон отдельно от описанных выше способов «отъема денег» у потребителя работать не может. Для выплаты гонорара должен образоваться фонд. Например, через метод «автодороги», «рекламы», «племенного быка» и т. п. У литераторов, например, этот фонд образуется методом «племенного быка», так как книга пока еще обладает товарными свойствами.

«Вал». Затраты на разработку символа измеряются и компенсируются производителю независимо от эффекта. Так «планируется» разработка ПО в наших НИИ. Эти методы достаточно раскритикованы и в «обычной» экономике.

«Степень». Оценивается деятельность производителя символов, и ему присваивается степень (звание), дающая некоторые льготы в потреблении товаров. Общество дает «остепененному» возможность и далее производить символы, не думая о хлебе насущном. Эффективность метода из-за подавления нового остепененными монополистами в области идей очень низка. Возможно, его эффективность можно повысить, вводя более широкие элементы демократии в общественные институты, реализующие этот метод.

«Приз». Система конкурсов, фестивалей, комитетов по премиям и т. п. В области программирования могла бы применяться гораздо шире, но, как и «гонорар», требует фондов, получаемых из других источников. К тому же, как и «степень», требует широкого демократизма. Именно институты общественного самоуправления придают товарам свойство «общественности» распределения, что и сглаживает основное противоречие.

Предложения применительно к программному продукту. Наберусь смелости внести предложения по некоторому сглаживанию основного противоречия знакообмена в процессе производства и потребления ПО широкого применения для универсальных ЭВМ общего назначения в условиях нашей экономики. Неблагодарное это занятие заранее обречено на неуспех, но, по моему мнению, существуют менее обременительные для общества способы сглаживать основное противоречие экономики информатики, чем предлагаемые классические способы информацию «сторожить».

Сделаю и еще одну оговорку. Эти мои предложения, и без того на грани здравого смысла, совершенно лишены его, если начать их внедрять, когда обычная экономика не сумеет еще развернуться в условиях полного, настоящего хозрасчета.

1. Устанавливается налог на производство средств вычислительной техники («Автодорога»), и за счет этого источника образуется фонд, из которого будет финансироваться все производство ПО широкого применения. Назовем этот фонд «экономическим фондом программирования» (ЭФП).

Часть этого фонда возвращается производителю ЭВМ, чтобы он мог на свой страх и риск финансировать разработчиков ПО, осуществляя профессиональный и заинтересованный контроль качества разработки. Другая часть распределяется по производителям ПО через соответствующие общественные организации (см. п. 5). Самое главное — не допустить, чтобы этот фонд прибрало к рукам какое-нибудь ведомство. Вообще, **от монополистов символьная экономика страдает в первую очередь и гораздо сильнее, чем товарная. Вот почему программирующая общественность должна самым решительным образом выступать против попыток монополизировать производство и эксплуатацию вычислительной техники и программного обеспечения.**

От искусства управления этим фондом прежде всего и будет зависеть будущее нашего программирования.

2. Программный продукт продается в магазинах по цене носителя плюс цена копирова-

ния. Это же относится к прилагаемой эксплуатационной документации, если таковая имеется.

За каждый проданный экземпляр разработчик и торговля субсидируются из ЭФП. Естественно, что максимальный размер субсидии должен быть сравним с ценой комплекта поставки. Иначе разработчик непопулярного ПО будет сам скупать свои изделия.

В большинстве случаев такое возмещение затрат на разработку окажется достаточным, так как стоимость носителей пока (к счастью для производителей символов) высока.

3. Применяются надбавки к цене на новую продукцию («мода»), устанавливаемые разработчиком. Последний присваивает дополнительную прибыль. Этот метод следует применять прежде всего для ПО, являющегося средством производства. Например, для экспертных систем и баз знаний.

4. Юридические запреты на копирование отсутствуют. Технические ухищрения, затрудняющие такое копирование, запрещаются, так как это не приведет к желаемой цели, а лишь создаст условия для подпольного бизнеса.

Однако не исключается защита от плагиата, т. е. от копирования с целью продажи и просовокупления к другому ПО.

5. Создается несколько (во всяком случае, более одного) независимых друг от друга общественных органов (общества потребителей ПО), которым передается часть ЭФП. Общества имеют свои выборные органы, комитеты по проведению конкурсов, фестивалей, опросов общественного мнения, издательства, исследовательские учреждения.

Общества объявляют конкурсы на разработку тех или иных программных продуктов и заключают договоры с кооперативами программистов или государственными предприятиями. Общества кредитуют исполнителей, частично несут экономическую ответственность за ошибки в ПО (страхование), участвуют в финансировании разработок экспертных систем, баз данных и знаний.

Общества имеют свои фонды программ, свои «фирменные» магазины, свои нормоконтролирующие органы.

6. Наряду с продажей можно использовать и прокат ПО.

7. Разработчик независимо от продажи собственно ПО может еще продавать и услуги на сопровождение. Вводятся нормы на сопровождение на правах подзаконного акта. Они устанавливают права и обязанности производителя и потребителя ПО. Например, обязательность поставки новых версий, извещений об ошибках, исправлений ошибок, неустойки. Цена таких услуг устанавливается производителем.

Выводы. 1. Попытки приравнять программный продукт к изделию производства (которое можно продавать) только на том основании, что и его производят, основано на смешении понятий разработки и изготовления, с одной стороны, и производства и потребления, с другой.

2. Противоречие между индивидуальным характером производства и общественным характером потребления информации неразрешимо в рамках всех имеющихся «товарных» экономических формаций. Эти формации перестанут существовать при смене характера экономики с товарного на информационный.

3. Понятие стоимости как общественно-необходимых затрат для каждого отдельного экземпляра информационного изделия лишено смысла.

4. Попытки сгладить указанное противоречие в условиях существующих формаций не приведут к цели. Наиболее действенный метод сглаживания — привлечение к распределительным функциям демократических институтов, создающих и увеличивающих «общественный» характер потребления.

5. Имущественное информационное право, основанное на запрете свободного распространения информационных изделий, крайне неэффективно и в основном вредно.

Телефон 234-58-74, Ленинград.

ЛИТЕРАТУРА

1. Электроника.— 1987.— № 11.— С. 84.
2. Электроника.— 1987.— № 7.— С. 82.
3. Беляков Б. Н., Зюляркин Д. Д., Подвальный А. М. и др. Производительные силы и производственные отношения в сфере производства программного продукта // Микропроцессорные средства и системы.— 1987.— № 2.— С. 3—6.
4. Маркс К. К критике политической экономии.— М.: Политиздат, 1984.
5. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации.— М.: Наука, 1984.
6. Эрнст Л. К. К высокому пределу // Химия и жизнь.— 1987.— № 9.— С. 2—8.
7. Электроника.— 1987.— № 2.— С. 5—6.

Статья поступила 20.03.88

УДК 681.324

В. В. Соловьев, В. В. Крывенко

ПРОГРАММНО-АППАРАТНАЯ ДОРАБОТКА ДВК3

Персональные ЭВМ ДВК3, работающие с ОС РАФОС, имеют ограниченные ресурсы для обмена информацией с другими машинами или устройствами. Разработана сервисная программно-аппаратная система для ПЭВМ типа ДВК3, которая позволяет:

осуществлять обмен информацией в виде файлов между дискетами и кассетами бытового магнитофона;

вводить программы с устройств внешней памяти, ретранслировать и записывать на дискету в форме файла типа MAC;

разрабатывать программы при помощи стандартных средств ОС и передавать их на объект с микропроцессорным управлением через магнитофон или микросхемы РПЗУ. В частности, возможна разработка новых или доработка (с предварительной ретрансляцией) существующих программ для ПЭВМ «Электроника БК-0010».

В систему входят:

модуль стыковки, выполненный в виде стандартной полуплаты, на которой реализованы интерфейсы ИРПР, НГМД, ИРПС для связи ПЭВМ с КНМЛ и ГМД-7012;

модуль программатора и программа обслуживания программатора, позволяющие программировать микросхемы РПЗУ;

две сервисные программы RETR и OBMI.

Программа RETR имеет несколько режимов работы, что позволяет загружать программные модули в ОЗУ из нескольких устройств внешней памяти в различных форматах записи, просматривать память в режиме экранной ретрансляции, получать на дискете исходный вид любой части ОЗУ в виде, пригодном для трансляции, либо редактирования.

Программа OBMI поддерживает обмен информацией между устройствами внешней памяти в нескольких форматах записи.

Основное назначение доработки — использование ДВК3 в качестве АРМ для подготовки и отладки ПО для управляющей бездисковой микропроцессорной системы, использующей команды микроЭВМ «Электроника 60», а также для документирования разработанных ранее программ в машинных кодах. Возможна организация архива рабочих программ на кассетах бытового магнитофона.

Телефон 59-82-60, Запорожье

Сообщение поступило 15.03.88

УДК 681.3.06

А. Д. Ермолаев, Н. П. Данилов

ПРОГРАММА ДЛЯ ВОССТАНОВЛЕНИЯ ИНФОРМАЦИИ, ЗАПИСАННОЙ НА АВАРИЙНЫХ ГИБКИХ МАГНИТНЫХ ДИСКАХ

Наиболее серьезным сбоем при выполнении операций ввода-вывода при работе с накопителями на гибких магнитных дисках (ГМД) в ОС ISIS-II и ДОС 1800 является так называемая ошибка № 24: обращение к аварийному блоку ГМД. При этом супервизор ОС прерывает операции ввода-вывода, что может стать причиной потери информации. Программа ERR 24 предназначена для полного или частичного восстановления информации, хранящейся в файле, обращение к которому с помощью обычных средств ОС приводит к ошибке № 24. Это особенно важно для файлов, несущих символьную информацию. Работа программы сопровождается сообщениями о физической локализации (номера сектора и дорожки) и характере (блок оглавления, указательный или информационный) аварийного блока. При этом степень потери информации зависит от характера аварийного блока. Программа имеет простое обращение, формат которого аналогичен вызову команды COPY в упомянутых выше ОС. Данная программа выгодно отличается от системной программы GANEF, так как в общем случае восстановление файла с помощью последней трудоемко и требует понимания файловой структуры ГМД. Алгоритм ERR 24 основан на комбинации операции ввода-вывода на физическом и системном уровнях.

101000, Москва, Хоромный тупик 2, ВНИИЭ,
тел. 924-96-59

Сообщение поступило 15.01.88

УДК 681.3.06

В. В. Семенов

ПРОГРАММА МНОГОМЕРНОГО РЕГРЕССИОННОГО АНАЛИЗА ДЛЯ МИКРОЭВМ

Программа многомерного регрессионного анализа (МРА) предназначена для построения линейных коэффициентов и сводимых к ним регрессионных моделей методом наименьших квадратов. Программа реализована на языке Паскаль и функционирует на микроЭВМ типа «Электроника 60» и совместимых с ней в ОС ФОДОС, РАФОС, ОС ДВК.

Организация диалога с помощью многоуровневых меню существенно упрощает освоение программы. Внутреннее представление исходных данных осуществляется в виде двумерного массива вещественных чисел, строки которого соответствуют наблюдениям, а колонки — факторам. Для максимального увеличения объема исходных данных программа МРА имеет оверлейную структуру, что позволяет при объеме оперативной памяти 28 Кслов анализировать до 200 наблюдений по 20 факторам. При необходимости можно увеличить максимальное число наблюдений (факторов), уменьшив число факторов (наблюдений).

Исходные данные вводятся в программу МРА из символьных файлов, подготовленных любым стандартным редактором текста, и файлов данных (рабочих), подготовленных программно.

Данные из информационных систем и баз данных можно конвертировать во внутренний формат рабочего файла, что позволяет гибко настраивать программу МРА на различные приложения. Для удобства анализа разнородных наблюдений ввод данных в программу МРА допустим из нескольких различных файлов, причем новые данные дозаписываются в конце двумерного массива.

Программа МРА включает статистические процедуры, позволяющие вычислять и выводить на терминал и печатающее устройство оценки следующих параметров регрессионной модели:

математических ожиданий, дисперсий, среднеквадратичных отклонений (СКВО) факторов;

коэффициент регрессии, их СКВО, частных критериев Фишера по каждому фактору;

адекватности модели в целом (коэффициент множественной корреляции, дисперсионное отношение, остаточная дисперсия).

Для выявления неоднородности исходных данных и неадекватности модели используются нормированные графики остатков, которые выводятся на печатающее устройство. Предусмотрен также вывод информационной матрицы, матрицы ошибок, корреляционной матрицы, что позволяет заметить мультиколлинеарность регрессоров, приводящую к неустойчивости оценок регрессионной модели, численной неустойчивости процедур оценивания, сильной взаимной корреляции коэффициентов регрессионной модели и увеличению их дисперсии.

Влияние мультиколлинеарности уменьшается благодаря использованию методов предварительного центрирования независимых переменных и регуляризации информационной матрицы. Для обращения информационной матрицы используется метод треугольного разложения (Холецкого), весьма устойчивый к плохо обусловленным положительно определенным матрицам, к которым относится рассматриваемая задача.

Неадекватность полученной на каком-либо этапе модели устраняется с помощью широкого набора процедур логического преобразования данных, позволяющего удалять и копировать факторы, проводить функциональные преобразования факторов вида 1/X, X**2, X**3, X**4, Sin X, Cos X, exp X, ln X, abs X, изменять масштаб факторов, получать парные произведения факторов.

Интерактивный режим работы программы МРА дает возможность эффективно осуществлять итерационный процесс построения адекватной модели регрессии, многократно повторяя цикл: оценка коэффициентов регрессии, проверка адекватности модели, логическое преобразование модели.

Программа МРА работает на микроЭВМ «Электроника 60» с объемом оперативной памяти не менее 28 Кслов, оснащенной дисплеем, накопителем на ГМД, печатающим устройством или ПЭВМ ДВК2, ДВК3. Загрузочный модуль программы МРА занимает на дискете 82 блока.

Выполняются следующие работы по развитию программы МРА:

«перегон» программы на ПЭВМ «Электроника 85», что позволит значительно увеличить объем анализируемых данных;

использование операций с двойной точностью при обращении информационной матрицы;

предоставление пользователю возможности определять точку в факторном пространстве, проводить наблюдения (эксперимент) с минимизацией обобщенной дисперсии.

Телефон: 266-26-88, Ленинград

Сообщение поступило 25.11.88

ПАМЯТКА АВТОРУ

Оформление рукописи

1. Объем статей (включая аннотацию, рисунки, выводы, список литературы) не должен превышать:

25 страниц машинописного текста для обзорной статьи;

15 страниц для статей о конкретных аппаратно-программных комплексах, устройствах и системах (включая фотографии внешнего вида, технические характеристики, тексты избранных подпрограмм);

10 страниц для статей концептуального характера.

3 страницы для рекламного материала или информационного сообщения.

2. Рукопись представляется в двух экземплярах, напечатанных через два интервала на одной стороне листа стандартного размера.

3. Используемую литературу необходимо давать общим списком в конце статьи в последовательности, соответствующей упоминанию источников в тексте.

4. Рисунки должны быть выполнены тушью с соблюдением ГОСТов.

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

6. К статье должно быть приложено направление учреждения, в котором выполнена работа, со всеми необходимыми сопроводительными документами (акты, справки и т. д.), а также сведения об авторах, содержащие их адреса и номера телефонов (служебные и домашние).

7. Перед названием статьи следует указывать индекс согласно универсальной десятичной классификации (УДК).

РОБОТОТЕХНИЧЕСКИЕ КОМПЛЕКСЫ

УДК 681.325

В. Т. Сайкин, И. Г. Артюх, И. И. Карпов, В. А. Юданов

УЧЕБНЫЙ РОБОТ

Ныне, как никогда, требуется, чтобы средняя и высшая школы выпускали из своих стен молодых людей с широким кругозором, идейно и нравственно воспитанных, с глубокими и прочными знаниями, способных овладеть достижениями научно-технического прогресса.

Решить эти задачи возможно лишь при целенаправленном сочетании как коренного улучшения экономической базы народного образования, так и идеологии самого учебного процесса.

Экономическая база общеобразовательной и профессиональной школ Москвы из года в год возрастает.

С 1984 по 1987 г. на содержание учреждений народного образования столицы, укрепление их материальной базы выделено из бюджета свыше 1,6 млрд. рублей. Значительно возросло строительство новых школ. Только за последние два года их было введено 65 на 73,5 тыс. мест, причем в этот период началось строительство школ по усовершенствованным проектам.

Однако коренного улучшения народного образования в столице, как и вообще в стране, не произошло. Об этом говорилось и на Пленуме Центрального Комитета партии, и на Пленуме Московского городского комитета.

Одна из основных причин такого положения — оторванность учебного процесса от реальной жизни, низкая техническая оснащенность учебных заведений, абстрактность в преподавании многих предметов, приводящая к пассивному восприятию учащимися.

Чтобы поправить положение дел, разработана целевая комплексная программа «Народное образование», которая предусматривает привлечение ведущих министерств, ведомств, особенно машиностроительного профиля к разработке и выпуску целого комплекса технических новинок для школ, решающих дидактические и организационно-педагогические задачи, ускоряющие адаптацию учащихся к современной технике.

Примером нового подхода к внедрению научно-технических средств в учебный процесс могут служить подготовленные НПО МЭП образцы учебных робототехнических комплексов и гидравлических стендов, получивших высокую оценку специалистов. Уже в 1988 г. около 70 экземпляров этих агрегатов поставлены для экспериментальной работы в школы Москвы, а с текущего года будет освоено их серийное производство.

Интенсивное внедрение вычислительной техники в нашу повседневную жизнь требует умения и навыков с помощью ЭВМ составлять программы управления различным производственным и бытовым оборудованием. Чтобы научить писать управляющие программы, необходимо иметь периферийные устройства, которые могли бы имитировать работу современного оборудования. Эти устройства могут быть простыми в изготовлении и безопасными в эксплуатации. Достаточно, чтобы они отображали только основные свойства моделируемого объекта. Такие простые модели будут познавательными игрушками, однако принесут большую пользу в создании условий для активного обучения [1—4].

В статье рассматривается новая модель учебного оборудования для трудового обучения учащихся общеобразовательных школ. Работы по изготовлению учебного робототехнического комплекса были начаты по инициативе Моссовета летом 1987 г. Комплекс предназначен для организации наглядного обучения основам программирования, управления робототехническими устройствами и знакомства с элементами гибких производственных систем. С его помощью моделируется работа грузоподъемных устройств, автоматизированных транспортных систем, металлообрабатывающих станков (сверлильного, фрезерного, гравировального) и станков типа «обрабатывающий центр». Комплекс состоит из манипулятора, блока сопряжения и подключается к ДВК2, ДВК3 через плату И2.

Манипулятор имеет семь степеней свободы (пять степеней подвижности), тип привода — электротехнический, точность движений — 0,5 мм, грузоподъемность — 0,2 кг, рабочая зона — два куба со стороной, равной 26 см.

Его кинематическая схема приведена на рис. 1, где символами DX, DY, DZ и стрелками обозначены крепления датчиков исходного (начального) положения манипулятора и датчиков относительных линейных перемещений соответственно вдоль осей OX, OY, OZ. Манипулятор имеет возможность поворáчивать схват относительно оси OY на 90° и разворачиваться на 90° относительно вертикальной оси (степени свободы W и F соответственно). Места установки датчиков исходного положения для этих степеней подвижности — DW и DF. Состояния «открыт-закрыт» схвата деталей и «включен-выключен» инструментальных блоков определяют R и S.

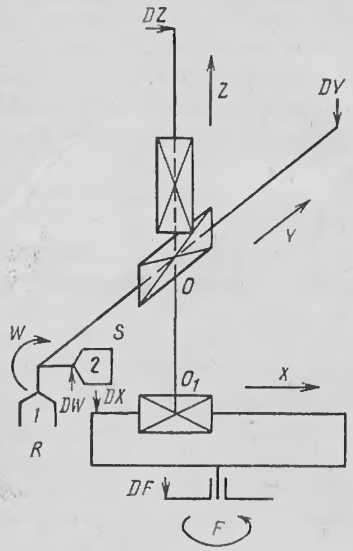


Рис. 1. Кинематическая схема манипулятора:
1 — устройство захвата для деталей и заготовок, 2 — устройство захвата для инструмента

Конструкция манипулятора (рис. 2) содержит основание, на котором установлено устройство, несущее три взаимно перпендикулярных модуля, каждый из которых состоит из трапецеидальной винтовой пары и двух цилиндрических направляющих (поз. 2). Привод на винт осуществляется от электродвигателя постоянного тока со встроенным редуктором через жесткую муфту (поз. 3). Вращение закрепленного на ходовом винте электродвигателя преобразуется в движение каретки (поз. 4), которая прикреплена к винтовой гайке. Число оборотов винтовой пары контролируется фотоимпульсными датчиками, а исходное нулевое положение устанавливается по аналогичным или магнитоуправляемым датчикам.

Исполнительный механизм состоит из поворотной головки (поз. 6), содержащей схваты деталей-заготовок (поз. 7) и инструментальных блоков (поз. 8).

Для перемещения, поворота и захвата детали или инструмента на электродвигатели по управляющему сигналу подается напряжение от ЭВМ. В процессе обработки управляющего сигнала в ЭВМ через устройство сопряжения поступают информационные сигналы с датчиков (поз. 5) о числе оборотов винта (на один оборот приходится восемь импульсов с фотоимпульсного датчика). Поворот головки осуществляется реверсированием двигателя до момента получения информационных сигналов от конечных выключателей (поз. 9).

В зоне действия манипулятора имеются магазин инструментальных блоков (рис. 3, поз. 2), два стеллажа для хранения деталей и заготовок и позиция обработки.

Предусмотрена возможность автоматической смены инструментального блока. В магазине инструментов их может быть не более трех (не принципиальное конструктивное ограничение). Каждый инструментальный блок содержит встроенный электропривод и цанговый зажим режущего инструмента. При использовании пишущих узлов в качестве инструмента манипулятор будет работать как графопостроитель.

Для управления манипулятором разработан пакет программ: программы нижнего уровня написаны на ассемблере и осуществляют обмен информацией между ЭВМ и блоком сопряжения; программы верхнего уровня — на БЕЙСИКе и реализуют алгоритм управления манипулятором, что позволяет учащемуся самостоятельно их модифицировать (обращение из верхнего уровня в нижний происходит с помощью оператора CALL).

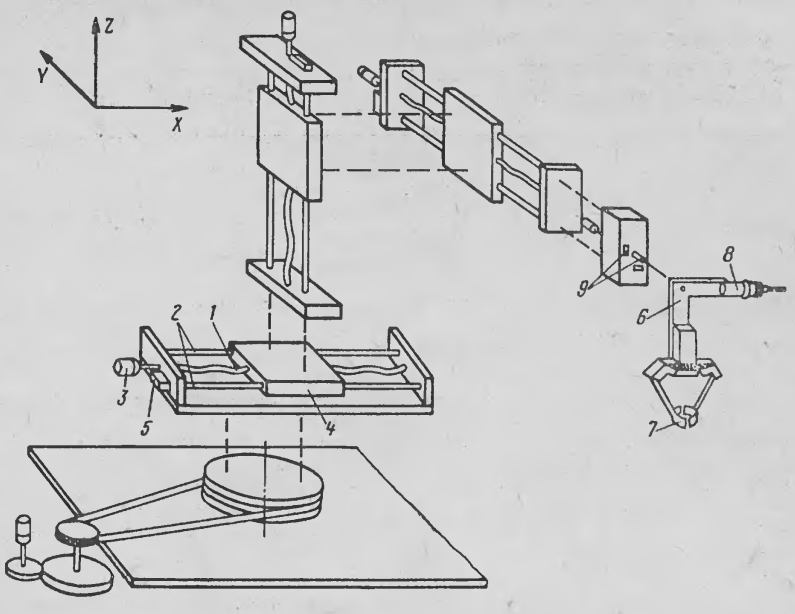


Рис. 2. Конструкция манипулятора

Обращение CALL BEGIN вызывает перемещение манипулятора в исходное положение, а CALL MOVE (C, X, Y, Z) — перемещение, определяемое значением переменной C и неотрицательными значениями переменных X, Y, Z. Для вычисления C заданного перемещения необходимо выбрать требуемые действия (табл. 1) и просуммировать соответствующие константы (например, $C=2^0+2^3=9$ означает одновременное перемещение манипулятора в положительном направлении оси OX на X шагов и в отрицательном по оси OY на Y шагов; перемещение на один шаг по любой координате равно 0,5 мм).

Если при суммировании выбраны исключающие друг друга действия, то на экран выдается сообщение об ошибке и выполнение программы прерывается. Для каждого перемещения программно пересчитываются абсолютные координаты X, Y, Z нового положения робота.

Таблица 1

Кодировка движений робота

Значение константы	Выполняемое действие
2^0	Движение по X в положительном направлении
2^1	Движение по X в отрицательном направлении
2^2	Движение по Y в положительном направлении
2^3	Движение по Y в отрицательном направлении
2^4	Движение по Z в отрицательном направлении
2^5	Движение по Z в положительном направлении
2^6	Вращение F (по часовой стрелке)
2^7	Вращение F (против часовой стрелки)
2^8	Вращение W (по часовой стрелке)
2^9	Вращение W (против часовой стрелки)
2^{10}	Схват закрыть
2^{11}	Подать напряжение к инструменту, находящемуся в кассете

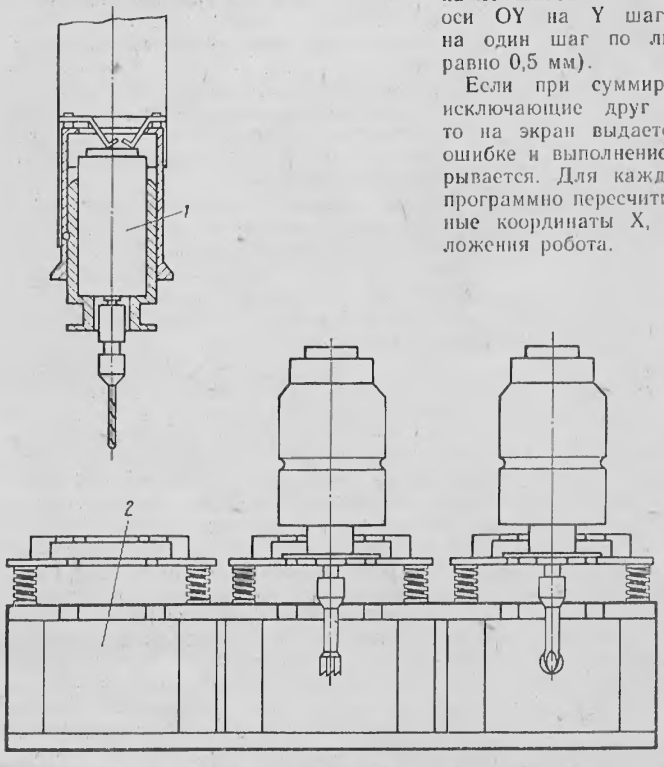


Рис. 3. Схват инструментальных головок и магазин

Таблица 2

Операторы и команды Т-языка

Операторы	Команды
BEGN	Установка манипулятора в исходное положение
MOVE X, Y, Z	Перемещение схвата в точку с абсолютными координатами X, Y, Z Примеры MOVE 100,, 20 MOVE ,,—50
MOVD DX, DY, DZ	Перемещение схвата в приращениях DX, DY, DZ к текущим координатам
MOVP DX, DY	Перемещение пера в приращениях DX, DY к текущим координатам
CRL X, Y, R	Построение окружности радиуса R с центром в точке (X, Y)
PROC ⟨ИМЯ⟩ ⟨ОПЕРАТОРЫ⟩ Т-ЯЗЫКА	Процедура Пример PROC SQ MOVP50,0 MOVP0,50 MOVP—50,0 MOVP0,—50
END	END Процедура SQ рисует квадрат 50×50, левый нижний угол которого находится в текущей точке
CLPR ⟨ИМЯ⟩	Обращение к процедуре ⟨ИМЯ⟩ Пример: CLPR SQ
PENU PEND	Пишущий узел поднять Пишущий узел опустить
TEXT X, Y, H, ⟨ТЕКСТ⟩	Пишет текст из букв высотой H, левый нижний угол первой буквы имеет абсолютные координаты X, Y Пример: TEXT 10, 20, 15, я рисую — значит, я существую
TAKE n PUT	Захват инструментальной головки с номером n Установка инструментальной головки в магазин
REPL n, ⟨ОПЕРАТОР⟩	Заменить оператор с номером n новым оператором Например: REPL 5, MOVE 0,, 15 Заменить оператор с номером 5 в создаваемой программе на оператор MOVE
SAVE ⟨ИМЯ ФАЙЛА⟩	Запомнить созданную программу в файле с указанным именем
READ ⟨ИМЯ ФАЙЛА⟩	Считать из файла с указанным именем текст в качестве текста текущей программы
EXEC TNTR	Выполнить созданную программу Перейти в режим непосредственного исполнения (после их ввода с клавиатуры)
CONT	Вернуться в режим занесения операторов в оперативную память

Программное обеспечение верхнего уровня разделено на подпрограммы, объединяемые программой «Монитор», которая организует работу комплекса в целом: выдает на экран перечень рабочих режимов, реализует диалог со школьником и вызывает переход в выбранный режим.

Предусмотрены следующие режимы работы: обучение робота, исполнение, программирование, имитация, рисование.

В режиме обучения школьник с помощью клавиатуры управляет перемещением манипулятора, выбирая требуемую траекторию. До начала движения задается ось, вдоль которой происходит перемещение, его дискретность (однократное нажатие клавиши) и направление перемещений (клавиши <0> или <1>). Постепенно изменяя шаг, можно привести схват в требуемое положение. От клавиатуры можно управлять не только перемещениями манипулятора по координатам X, Y, Z, но и положением схвата, включением-выключением инструмента (S) и схвата (R), поворотом манипулятора вокруг собственной оси (W). При нажатой клавише <, > текущее состояние манипулятора (переменные X, Y, Z, W, F, R, S) запоминается в оперативной памяти. Последнему состоянию последовательности работ (клавиша <M>) присваивается имя, которое заносится в каталог работ, выполняемых роботом. При воспроизведении траектории происходит ее оптимизация с помощью совмещения перемещений по нескольким степеням подвижности. Таким образом, из траектории, по которой двигался робот, удаляются все ненужные перемещения. Пример организации диалога в режиме обучения приведен на рис. 4.

В режиме исполнения с дискеты считывается и выводится на экран список работ, выполняемых роботом. Если в процессе выполнения запрограммированного движения необходимо ввести новые параметры, то можно сделать вставку для организации диалога в соответствующее место программы. В матобеспечении робота приводится программа (на языке БЕЙСИК) для моделирования рабочего цикла.

Для составления управляющих программ обычно пользуются ориентированными на такие задачи языками программирования, поэтому в ПО включен Т-язык для программирования движений робота (табл. 2). Предложения Т-языка состоят из: операторов, задающих движения робота, и команд, обрабатывающих текст программы (распечатывают программу на экране, записывают ее во внешний файл).

В программное обеспечение включена программа, имитирующая движения робота на экране дисплея, которая позволяет школьнику наблюдать перемещения инструмента от точки к точке на экране дисплея (при ошибках в программе манипулятор выполняет непредусмотренные действия, которые необходимо устранить). Наблюдение может вестись в реальном темпе показа движения, ускоренном или в стоп-кадре.

Школьник самостоятельно может заставить манипулятор рисовать на горизонтальной поверхности отрезки прямых линий, окружности, писать в заданном месте текст, состоящий из букв заданного размера.

Первые занятия с роботом начались в кабинете информатики московской школы № 693. Десятиклассников знакомят с пакетом программ, предлагают задачи, для решения которых необходимо внести изменения в эти программы. Школьники наглядно осознают возможность применения на практике знаний, полученных на уроках информатики.

В создании описанного учебного робота (см. 1-ю стр. обложки) принимали участие инженеры: Витвицкая Л. В., Гущина Г. И., Ермолаев С. Н., Кузьменко В. Ф., Лебедев И. Б., Мацук А. К., Михайлов В. А., Монахова Л. С., Никулушкина О. Н., Попов С. Е., Пчелинцев В. А., Рого-

жин Н. М., Сараева В. В., Уманский С. Д.

Телефон 332-92-36, Москва

ЛИТЕРАТУРА

1. Клейман Г. М. Школы будущего: компьютеры в процессе обучения.— М.: Радио и связь, 1987.
2. Марш П., Александер И., Барнетт П. и др. Не счесть у робота профессий / Под ред. В. С. Гурфинкеля.— М.: Мир, 1987.
3. Логинов В., Высокодворский В., Казанский Ю. Робот для школьника // Информатика и образование.— 1987.— № 6.— С. 115—116.
4. Дуванов А., Зайдельман Я., Первин Ю. Роботоландия // Информатика и образование.— 1988.— № 1.— С. 109—116.

Статья поступила 19.05.88

```
ЕСЛИ ВЫ ВПЕРВЫЕ УПРАВЛЯЕТЕ РОБОТОМ,  НАЖМИТЕ < Н >

                          < X > < Y > < Z >
                             < F > < W >
                          < , > < H > < M >

НАЖМИТЕ <ВР> И ОДНУ ИЗ КЛАВИШ

**********  НАЖАТА КЛАВИША  < X >  **********

ВВЕДИТЕ ШАГ ( 1 , 2 , 3 ,... , 25 )

     ШАГ = 5

< 0 > - ВЛЕВО , < 1 > - ВПРАВО

< ВК > - ЗАКОНЧИТЬ ДВИЖЕНИЕ

 ВАШ ХОД ?
РОБОТ ДВИЖЕТСЯ ВПРАВО
```

Рис. 4. Вывод информации на экран в режиме обучения робота

УДК 681.325

Макаров И. Н. и др.

УЧЕБНАЯ ГИБКАЯ ПРОИЗВОДСТВЕННАЯ СИСТЕМА*

Широкое внедрение в промышленность роботов и гибких производственных систем (ГПС) требует существенного расширения и повышения эффективности системы подготовки и переподготовки специалистов. Сейчас эта задача — общая для всех промышленных развитых стран [1, 2].

Наряду с тем, что многие промышленные фирмы (например, в США) самостоятельно организуют краткосрочные курсы от 100 дней до девяти месяцев для ускоренной подготовки и переподготовки инженеров и техников, активное участие в подготовке специалистов в области робототехники принимают вузы. В университетах и колледжах США действует 256 учебных программ по робототехнике. Именно в вузах сосредоточено более половины всех учебных лабораторий по робототехнике, которые оснащены специальными учебными (Microrobot, Rhino) и промышленными роботами (Puma, Asea) и некоторыми элементами технологического оборудования (поворотными столами, конвейерами).

Применение комплекса «Учебная ГПС» при подготовке студентов-робототехников позволяет:

изучить на практике будущему специалисту принципы построения, управления, а также технологию ГПС;

создать современную исследовательскую базу для самостоятельной практической научно-исследовательской работы в процессе обучения.

Структура аппаратных средств учебной ГПС

Учебная ГПС состоит из шести модулей (см. вкладку).

Схема управления аппаратными средствами учебной ГПС приведена на рис. 1.

ЭВМ верхнего уровня (МЕРА-60) управляет основными режимами работы ГПС и отдельных модулей. Их взаимодействие осуществляется с помощью многоканальной платы устройства последовательного обмена (УПО) (связь с транспортной тележкой и системами управления трех сборочных центров) и расширенного варианта платы параллельного обмена (И2, управления складов и включения механообработки).

Для повышения нагрузочной способности выходных каскадов И2 применен блок инвертирующих усилителей. Разряды 00...01 регистра команд и состояния используются для управления выдачей данных. Запись «1» в эти разряды разрешает вывод данных из младшего и старшего байтов выход-

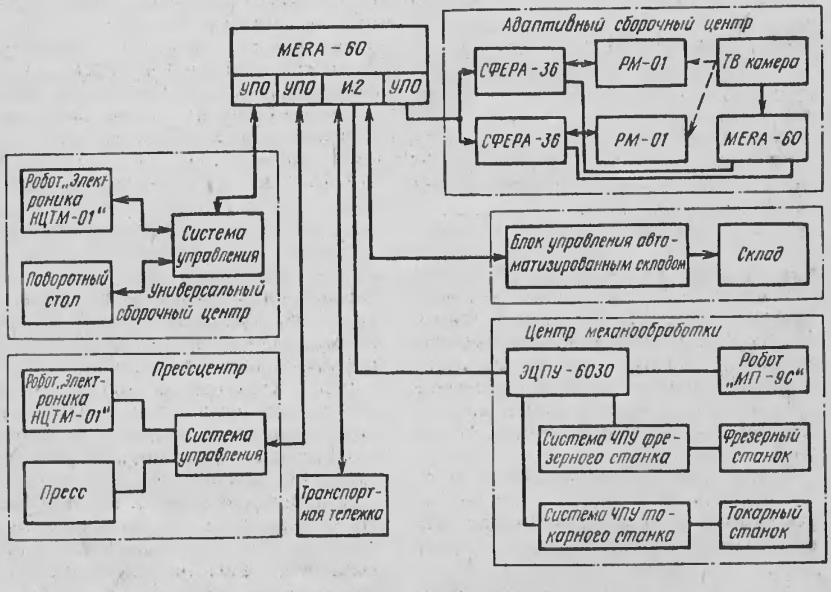


Рис. 1. Схема управления аппаратными средствами учебной ГПС

* Учебная ГПС разработана на кафедре «Проблемы управления» МИРЭА под руководством академика Макарова И. М. коллективом сотрудников: Лохиным В. М., Винокуровым К. Е., Штыковым А. В., Никольским А. Ф., Курдюмовым В. В., Костиным М. Ю., Объедковым А. Н., Романовым М. П., Вавером А. Е., Спиридоновым Г. Н., Тягуновым О. А.

ного слова соответственно. Управление контроллером склада осуществляется через старший байт выходного слова, а старший разряд младшего байта формирует сигнал включения центра механообработки.

Состав мини-ЭВМ МЕРА-60: ОЗУ объемом 256 К 16-разрядных слов, пять дисплеев (один системный), устройство печати, накопитель на гибких магнитных дисках ГМД-7012, два накопителя СМ5200, четырехканальное УПО, устройство параллельного обмена (И2), блок инвертирующих усилителей (центральный процессор — М6).

Операционная система ЭВМ верхнего уровня — RT-11.05 (Фодос 2) с многозадачным XM монитором, который поддерживает восемь заданий (фоновое, оперативное и шесть системных) и работу с расширенной памятью (256 Кслов).

Состав ЭВМ нижнего уровня: процессор К1801ВМ1 ПЗУ (8 К 16-разрядных слов), в котором записана программа управления роботом «Электроника НЦТМ-01» и технологическим оборудованием, две платы И2 для связи с блоком ключей и инверторов.

Управление ЭВМ в пультовом и в программном режимах осуществляется через УПО с помощью системного терминала.

Структура программных средств учебной ГПС

Состав ПО:

программа связи TEM организует оперативное задание на использование системного терминала ЭВМ верхнего уровня в качестве системного терминала нижнего уровня;

программа-загрузчик BOOT обеспечивает загрузку программы в память ЭВМ нижнего уровня;

четыре программы управления, формирующие системные задания для работы склада, универсального и специализированного сборочных центров, центра механообработки и организующие запросы на обслуживание соответствующих модулей ГПС транспортной тележкой;

системный отладчик ODT;

монитор системы вырабатывает фоновое задание, контролирующее общую работу системы;

пять драйверов (телеграфного канала связи с ЭВМ, управляющей пресс-центром ZV и универсальным сборочным центром YV; памяти для считывания информации в ОЗУ; ее интерпретации как устройства прямого доступа без каталога; управления транспортной тележкой; типа «почтовый ящик» для организации обмена информацией между заданиями).

Основные программные средства

Драйвер телеграфного канала YV (ZV) организует ввод-вывод информации по телеграфному каналу. Он имеет три точки входа: инициации и завершения ввода-вывода, обработки прерываний. Точка входа инициации ввода-вывода задается макрокомандой DRBEG и анализируется номер устройства: если он четный, драйверу передается число необходимых слов, в противном случае — число байтов. При отрицательном значении счетчика слов (байтов) производится операция вывода, при положительном — операция ввода. При значении, равном 0, устанавливается режим эмуляции, действующий до первого запроса на операцию ввода, при котором осуществляется связь системного терминала с ЭВМ нижнего уровня. При операции ввода принимаемый байт записывается в регистр данных системного терминала. Точка входа в подпрограмму обработки прерываний задается макрокомандой DRINT. После обмена информацией между регистрами данных и буферами ввода-вывода происходит коррекция счетчика слов (байтов) и, если он равен нулю, осуществляется переход на подпрограмму завершения операций ввода-вывода, которая запрещает прерывания и возвращает управление монитору.

Драйвер управления транспортной тележкой формирует команду управления.

Четыре старших разряда командного слова управления транспортной тележкой содержат информацию для контроллера тележки, остальные задают длину пути (время задержки).

Подпрограмма инициации дешифрирует командное слово. Если четыре старших разряда равны 0, управление передается подпрограмме обработки паузы, которая организует запрос на запрет подпрограммы завершения через заданный интервал времени (используется подпрограмма дешифрации следующей команды).

После дешифрации драйвер устанавливает в регистре данных соответствующие управляющие биты и разрешает движение тележки.

Программа обработки прерываний сравнивает текущий пройденный путь с заданным и при совпадении выдает команду останова. После обработки всех команд драйвер возвращает управление монитору.

Загрузчик BOOT открывает канал связи с ЭВМ нижнего уровня, инициирует работу первичного (аппаратного) загрузчика и заполняет с его помощью вторичный загрузчик LOAD. После этого он открывает один или несколько файлов в формате абсолютного загрузчика и переносит их в память ЭВМ нижнего уровня.

LOAD анализирует стартовый адрес программы: если он четный, вводит программу с этого адреса, в противном случае — ожидает загрузку следующего файла.

LAMP управляет датчиками склада. Ее входные аргументы — номера позиций, задающие положение штабелера по трем координатам. Подпрограмма вырабатывает управляющее слово, разрешает его передачу на контроллер склада и выдает сигнал переключения датчиков. После окончания работы вывод информации запрещается — возможность ложного срабатывания исключается.

DVIG управляет двигателями склада. Номер привода (от —3 до +3, где знак числа указывает на направление движения) является индексом входа в таблицу управляющих слов и масок для анализа состояния датчиков. Такое решение обеспечивает максимальную гибкость программирования режимов работы двигателей.

После полной остановки штабелера производится окончательная проверка точности заданной позиции.

Структура и принцип действия модулей учебной ГПС

В состав **центра механообработки** (ЦМО) входят: токарный (А-195EV) и фрезерный (А-200EV) станки с системами ЧПУ модели DIN 66025 производства фирмы Elettronica Veneta (Италия), цикловый робот МП-9С с пультом управления ЭЦПУ-30, накопитель деталей, порт приема с приемной кассетой.

Центральный процессор ЧПУ выполнен по схеме МП модели Z80 и имеет 150 команд, выполняемых за время от 1 до 4,5 мкс. ПЗУ EPROM (30 Кбайт) обеспечивает режимы автоматического и пошагового управлений. ОЗУ (16 байт) хранит промежуточные расчеты, необходимые для выполнения операций управления. Устройства ввода-вывода формируют управляющие команды и принимают необходимую информацию.

Пульт ЭЦПУ-6030 задает режим командного управления исполнительными элементами робота и зажимных устройств станков, а также включение стоек ЧПУ для работы в программном режиме. Первая рабочая операция — захват манипулятором заготовки из накопителя и ее установка в патрон токарного станка. После режима токарной обработки деталь переносится в зону механических тисков фрезерного станка и зажимается по команде. Команда включения фрезерного станка совмещена с циклом возвращения манипулятора в исходное положение.

Центр тяжести обрабатываемой детали на фрезерном станке находится за пределами тисков, поэтому после завершения режима фрезерования и разжатия тисков деталь по жолобу попадает в приемный лоток.

Очередная заготовка подается из накопителя во время обратного хода манипулятора, при котором он надавливает на поворотный рычаг толкателя, перемещающего заготовку из накопителя в зону захвата детали манипулятором.

Автоматизированный склад (АС) — металлическая конструкция для хранения кассет с готовыми деталями и заготовками, порта приема транспортной тележки, блока управления авто-

матизированным складом (БУАС) и штабелера.

В секции для хранения располагается не более 12 кассет (по три в ряд) размерами 300×300 мм (высота кассеты с деталями не должна превышать 220 мм).

Порт приема оборудован направляющими для кассет, которые обеспечивают их точное позиционирование и ориентацию для компенсации возможного отклонения транспортной тележки от программной траектории движения.

Штабелер работает в позиционном режиме и производит перезагрузку транспортной тележки как с нижней, так и с верхней приемных позиций. С помощью трех электромеханических приводов штабелер перемещается в горизонтальном, вертикальном и продольном направлениях.

Четыре точки позиционирования по горизонтальной оси соответствуют положению штабелера у порта приема транспортной тележки и у секции для хранения кассет, восемь точек позиционирования по вертикальной оси — расположению каретки ниже и выше каждого из четырех рядов секций для хранения кассет. Это позволяет вынимать и устанавливать кассеты в секции хранения. В продольном направлении подвижные виды каретки штабелера могут быть расположены в двух точках, соответствующих их втянутому и выдвинутому положениям.

На раме, каретке и вилах расположены оптические датчики, выходы которых соединяются с управляющей ЭВМ. Положение светодиодов, установленных в точках позиционирования, определяет координаты.

Для управления складом используются два типа команд (рис. 2). Первой подается команда включения светодиода, расположенного в желаемой точке позиционирования штабелера, затем команда включения соответствующего электродвигателя в нужном направлении, определяемом управляющей ЭВМ по известному исходному положению штабелера и координатам следующей точки позиционирования. После этой команды ЭВМ переходит в состояние опроса выходов оптических датчиков. Появление Лог. 0 означает нахождение штабелера в данный момент в заданной точке позиционирования. В этом случае ЭВМ выдает команду на выключение соответствующего двигателя.

Длина управляющей команды равна одному байту, при этом два старших разряда указывают тип выполняемой команды. Первые два бита команды управления состоянием светодиодов должны содержать 0, состоянием электродвигателей — 1, остальные шесть — информацию о текущем состоянии светодиодов или включенных электродвигателях и направлении их вращения.

По команде управления состоянием светодиодов ДК вырабатывает сигнал, разрешающий запись битов в регистр

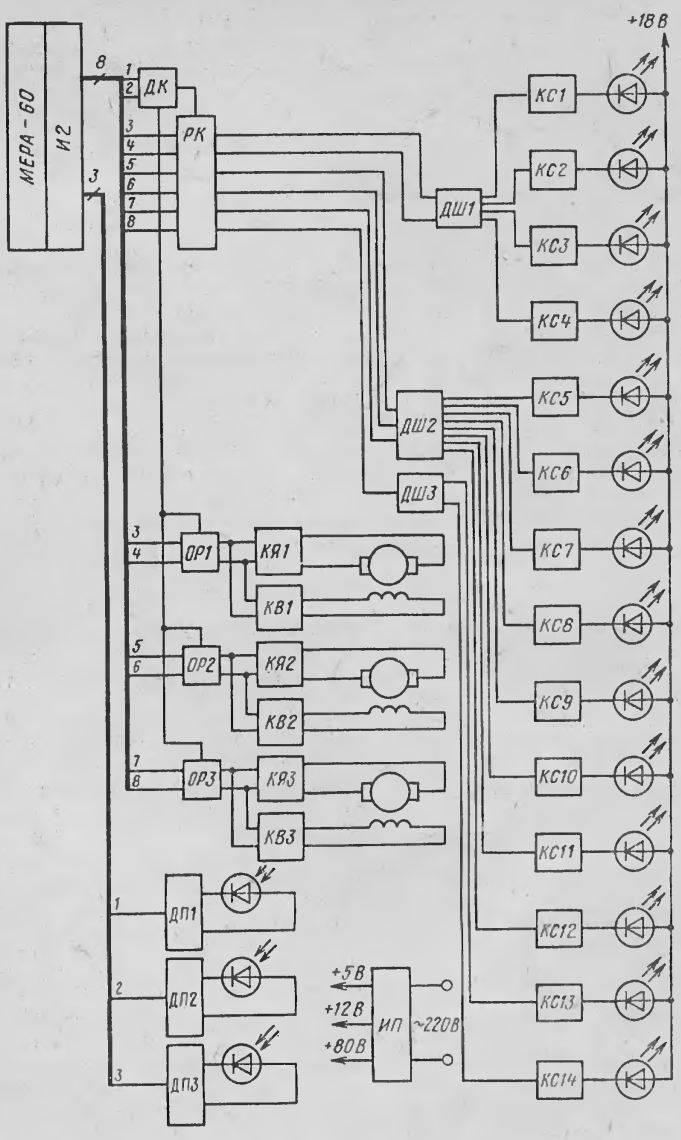


Рис. 2. Схема блока управления автоматизированным складом:
дешифратор кода команды — ДК; регистр хранения адреса включенного светодиода — РК; дешифраторы адреса светодиодов, задающие координаты точек позиционирования: в горизонтальном — ДШ1, вертикальном — ДШ2 и продольном — ДШ3 направлениях; ключи управления состоянием светодиодов — КС1, КС2, ... КС14; оптронная развязка от силовых цепей питания электродвигателей штабелера — ОР1...ОР3; ключи управления обмотками возбуждения — КВ1 ...КВ3 и якорными обмотками — КЯ1...КЯ3

хранения адреса зажженного светодиода, к выходу которого подключены дешифраторы, преобразующие двоичный код в позиционный, используемый для управления ключами КС1...КС14 (определяется состояние светодиодов по всем трем координатам).

При поступлении на вход блока управления автоматизированным складом (БУАС) команды управления электродвигателями ДК вырабатывает сигнал, разрешающий ее прохождение через оптронную развязку на КВ1...КВ3 и КЯ1...КЯ3. Данная команда не запоминается, поэтому можно реализовать как релейный, так и программно организованный время-импульсный режим управления двигателями и отключать приводы штабелера при неисправности управляющей ЭВМ.

По одной команде БУАС можно включить все три привода штабелера. Возможно одновременное движение в горизонтальном, вертикальном и продольном направлениях при втянутых вилах либо одновременное втягивание вил и вертикальное движение. Однако недопустимо движение в горизонтальном направлении при выдвинутых вилах. Правильность задания команд управления электродвигателями реализуется программой управляющей ЭВМ.

Универсальный сборочный центр (УСЦ) включает в себя (рис. 3):

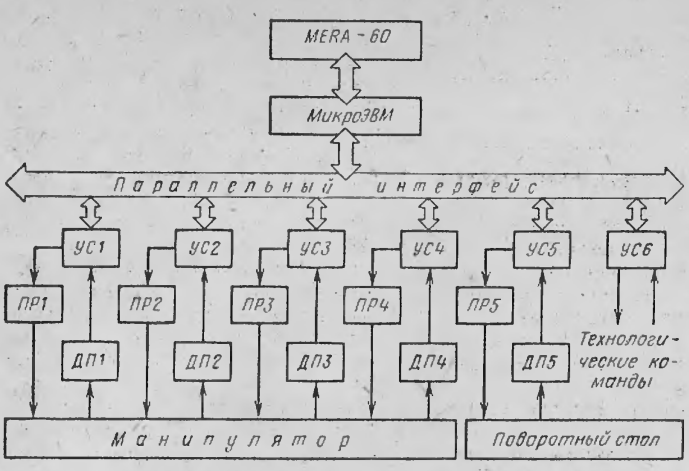


Рис. 3. Схема системы управления универсальным сборочным центром:

устройства сопряжения — УС1...УС6, преобразователи — ПР1...ПР5, датчики положения робота — ДП1...ДП5

робот «Электроника НЦТМ-01» [3], микроЭВМ «Электроника НЦ-80», поворотный стол, сборочную кассету, накопитель (фото 6 вкладки). Для удобства выполнения операций сборки серийный пневмозахват на роботе заменен на электрический с регулируемым усилием зажатия изделия.

дующим накоплением в программно реализованном счетчике.

Система управления поворотным столом осуществляется по схеме однополярного частотно-импульсного модулятора (ОЧИМ) с ПИД-регулятором [4, 5]. Применение ОЧИМ обеспечивает качественно высокие показатели,

Технические характеристики УСЦ

Число деталей и узлов, одновременно поступающих из накопителя, шт, не более	4
Число собираемых на кассете изделий, шт.	4
Габаритные размеры собираемых деталей и узлов, мм	80×80×50
Масса собираемых деталей и узлов, кг, не более	1
Габаритные размеры кассеты, мм	300×300×50
Диаметр поворотного стола, мм	700
Точность позиционирования поворотного стола, мм	0,36
Привод поворотного стола	ЭТА-2
Габаритные размеры УСЦ, мм	1550×1800×1250

Рассмотрим принцип работы УСЦ (рис. 3). Комплект деталей для одного собираемого изделия поступает из накопителя в ячейки поворотного стола, который, поворачиваясь на заданный угол, подает в рабочую зону робота упорядоченный комплект деталей. Робот перемещает детали одну за другой в место сборки на кассету; поворотный стол загружается из накопителя и осуществляется поворот новой партии деталей в рабочую зону робота, который собирает второе изделие на втором посадочном месте кассеты (пока кассета не будет полностью заполнена собранными изделиями).

В роботе «Электроника НЦТМ-01» трехпозиционное безгистерезисное реле с двумя уровнями питающего напряжения управляет двигателями и обеспечивает две скорости движения робота. Для улучшения качества движения используются электротормоза, вмонтированные в двигатели.

Для реализации плавного разгона и торможения при управлении поворотным столом использовано программное частотно-импульсное управление двигателем, а датчик положения выполнен по принципу счета импульсов с последнадежное управление при малых уровнях рассогласования, а также вибрационную линеаризацию нелинейности типа «зона нечувствительности» исполнительных двигателей.

Программы управления УСЦ написаны на языке Паскаль с использованием фрагментов ассемблера.

Специализированный сборочный центр (ССЦ) включает в себя: робот «Электроника НЦТМ-01М» с системой управления, вспомогательное оборудование для выполнения операций запрессовки, две кассеты с втулками и валами. Вал из кассеты робот помещает в кондуктор пресса, затем берет втулку и опускает в посадочное место кондуктора над валом. Далее робот захватывает фиксатор пресса, перемещает каретку с кондуктором — происходит операция запрессовки (вал вдавливается во втулку), затем возвращает каретку с кондуктором в исходное положение, фиксирует ее, вынимает готовое изделие из кондуктора и помещает в кассету готовых изделий. Система управления сборочным центром аналогична системе управления УСЦ.

Транспортная тележка (ТТ), используемая в учебной ГПС, разработана в Ленинградском механическом институте. Она имеет три ведущих колеса, расположенных в вершинах равностороннего треугольника. Сверху установлена платформа с жестко закрепленными нижними и верхними несущими вилами, на которых размещаются две кассеты с готовыми изделиями или заготовками.

Каждое из колес может вращаться относительно вертикальной и горизонтальной осей. Управление движением каждого колеса осуществляется с помощью двух двигателей постоянного тока типа ДПР-62 с планетарными редукторами.

Углы поворота вокруг вертикальной оси фиксированы датчиками угла с концевыми выключателями, потому возможно вращение тележки на месте и передвижение прямо и боком.

Вращение колес вокруг горизонтальных осей приводит к поступательному перемещению тележки (датчик, программирующий величину перемещения, расположен на ведущем колесе).

Напряжение питания и сигналы управления подаются на тележку по подвешенному гибкому кабелю.

В состав **адаптивного сборочного центра** (АСЦ) входят два робота РМ-01 со стойками управления «Сфера-36» и система технического зрения (СТЗ) на базе мини-ЭВМ «МЕРА-60». В АСЦ изделия собираются автоматически из деталей, исходное положение которых в рабочей зоне неизвестно. Рабочие зоны роботов пересекаются. Кассета с деталями и кассета, в которой будет происходить сборка изделия, устанавливаются в область перекрытия рабочих зон.

СТЗ позволяет определить расположение деталей в кассете и ее ориентацию для исключения возможности столкновения роботов в процессе сортировки деталей или сборки изделия.

Центральная ЭВМ выдает сигнал на разрешение работы АСЦ в момент доставки кассеты на сборочный центр. По этому сигналу один из роботов снимает кассеты с тележки и устанавливает их в позицию сборки (формируется сигнал о разгрузке транспортного робота). Далее сборочный центр работает автономно.

После сборки изделия мини-ЭВМ «МЕРА-60» формирует сигнал о необходимости перезагрузки сборочного центра (он переводится в режим ожидания). Перезагрузка осуществляется с помощью транспортного робота, процессы сортировки и сборки повторяются.

Комплекс исследовательских работ студентов на учебной ГПС

В учебную ГПС входят не только лабораторные работы по основным дисциплинам специальности «Робототехнические системы и комплексы», но и самостоятельная научно-исследовательская работа, курсовое и дипломное проектирование (см. таблицу).

Таблица

Комплекс исследовательских работ студентов на учебной ГПС

Модуль	Наименование дисциплины	Предмет исследования
АСЦ	ГПС, ТРП	Технология роботизированной сборки, принципы построения модулей сборочных ГПС
	СУРМ	Контурная система управления роботом РМ-01.
	ВУРТС	Архитектура вычислительного устройства «Сфера-36». Язык программирования VAL.
	ИУР	Принцип действия и характеристики СТЗ, алгоритмическое и программное обеспечение СТЗ.
	ПРТС	Динамика электропривода робота РМ-01
УСЦ	ГПС, ТРП	Технология роботизированной сборки, принцип действия модулей сборочных ГПС
	СУРМ	Позиционная система управления роботом «Электроника НЦТМ-01»
	ВУРТС	Архитектура вычислительного устройства «Электроника НЦТМ-30». Языки программирования (Паскаль, ассемблер и др.)
	ПРТС	Динамика электропривода робота «Электроника НЦТМ-01» и поворотного стола с частотно- и широтно- импульсной модуляцией
ССЦ	ГПС, ТРП	Технология роботизированной сборки, принцип действия модулей сборочных ГПС
	СУРМ	Позиционная система управления роботом «Электроника НЦТМ-01М»
	ПРТС	Динамика электропривода с широтно-импульсной модуляцией
ТТ	ПРТС	Динамика двухконтурной системы подчиненного регулирования и системы управления с широтно-импульсной модуляцией
	СУРМ	Структура, принцип действия, алгоритмическое и программное обеспечения системы управления ТТ
	ГПС	Транспортные средства ГПС
ЦМО	СУРМ	Цикловая система управления роботом МП-9С
	ПРТС	Электрические приводы токарного и фрезерного станков, пневматический привод робота
	ВУРТС	Архитектура и особенности программирования устройства ЭЦПУ-30
	ГПС, ТРП	Модуль механообработки с системой ЧПУ как элемент ГПС
АС	ГПС, ТРП	Конструкция и принцип действия типового склада ГПС, аппаратные и программные средства
	ПРТС	Динамика электропривода с релейным, частотно-импульсным и широтно-импульсным управлением
Учебная ГПС	ВС	Состав, принцип действия ГПС и ее отдельных модулей
	ВУРТС	Аппаратно-программные средства ГПС, программирование режимов работы ГПС
	МРГПС	Полунатурное моделирование ГПС и отдельных модулей
	ГПС	Принципы построения ГПС механообработки и сборки

Примечание: ВС — введение в специальность, ПРТС — приводы робототехнических систем, СУРМ — системы управления роботов и манипуляторов, ВУРТС — вычислительные устройства робототехнических систем, МРГПС — моделирование робототехнических систем и ГПС, ИУР — информационные устройства роботов, ГПС — гибкие производственные системы, ТРП — технология роботизированных производств.

Заключение

В данной статье авторы рассказали об одной из разработок МИРЭА в области учебных робототехнических комплексов. Кроме учебной ГПС введена в эксплуатацию и рекомендована комиссией Минвуза СССР к тиражированию для вузов страны экспериментальная учебная лаборатория «Системы управления роботов и манипуляторов» и разрабатывается новая версия учебной ГПС на базе малогабаритных роботов и станков с управлением от локальной сети IBM PC. Об этих новых разработках авторы предполагают рассказать в последующих публикациях.

117454, Москва, пр. Вернадского, 78, МИРЭА, кафедра «Проблемы управления», тел.: 434-93-67

ЛИТЕРАТУРА

1. Макаров И. М., Медведев В. С., Ющенко А. С. Подготовка специалистов по робототехнике // Вестник высшей школы.— 1987.— № 4.— С. 86—88.
2. Education and training in Robotics // Ed. T. M. Husband IFS. Ltd. UK.— Springer—Vertag, 1986.— P. 315.
3. Робототехника и гибкие автоматизированные производства. Гибкие автоматизированные производства в отраслях промышленности, кн. 7. Учебное пособие для втузов в 9-ти кн. / Под ред. И. М. Макарова.— М.: Высшая школа, 1986.
4. Робототехника и гибкие автоматизированные производства. Приводы робототехнических систем, кн. 7. Учебное пособие для втузов в 9-ти кн. / Под ред. И. М. Макарова.— М.: Высшая школа, 1986.
5. Лохин В. М., Мадыгулов Р. У., Романов М. П. Исследование движений в частотно-импульсных системах // Техническая кибернетика.— 1985.— № 4. (Тр. Известия АН СССР).

Статья поступила 12.10.88

УДК 658.512.011.56:621.9.06—529

С. В. Кузьмичев, А. М. Набутовский, В. П. Селезнев

СИСТЕМА ПОДГОТОВКИ ПРОГРАММ ДЛЯ ПРОГРАММИРУЕМЫХ КОНТРОЛЛЕРОВ

Для управления цикловыми станками и автоматическими линиями широко применяются программируемые контроллеры (ПК). В каждом — свой язык, позволяющий реализовать требуемый алгоритм. Основа всех языков — логические уравнения, написанные с помощью алгебры Буля (в языке ПК NS-915 Тесла — мнемокод, в ПК 32...ПК 248 МикроДАТ — релейно-контактная символика). Каждое уравнение алгебры Буля может однозначно транслироваться в набор команд языка конкретного контроллера. Эта работа осуществляется вручную, занимает много времени и требует высокой квалификации.

Для автоматизации подготовки программ для ПК были разработаны системы программ для ПК 32...ПК 248 и NS-915. Системы написаны на языке Турбо Паскаль и работают под управлением ОС СР/М (SCP-0006) на ПЭВМ ROBOTRON-1715 с объемом оперативной памяти не менее 48 Кбайт.

Работа происходит в полном диалоговом режиме с помощью многоуровневых меню.

Рассмотрим систему подготовки программ для ПК 32...ПК 248. После загрузки системы оператору предлагается главное меню, по которому выбирается один из режимов работы: выбор дисковода, считывание файла, создание таблицы имен, редактирование, задание маски обмена, документирование, запись-копирование файла, конец работы.

Программы могут редактироваться, копироваться, что позволяет быстро выпускать программы исполнений станков. Для удобства работы на дисплее индицируется каталог программ текущего дисковода. До начала работы необходимо «считать» файл для редактирования. При отсутствии имени в каталоге система организует новый файл.

Программирование осуществляется как в абсолютных адресах входов, выходов и внутренних маркеров ПК, так и в символических именах (режим «создание таблицы имен»). Режим допускает просматривать таблицу (рис. 1), удалять существующие символы, вводить новые с контролем двойных символов и адресов и проверкой на допустимость адреса ПК. Ввод символов возможен в любом порядке. В этом же режиме таблицу имен можно вывести на печать. В момент выхода в основное меню производится сортировка таблицы (символы располагаются по порядку адресов).

В режиме «Редактирование» на экране появляется релейно-контактная цепочка и оператору предлагается следующее меню функций: 1 — предыдущий, 2 — следующий, 3 — начало, 4 — конец, 5 — редактирование, 6 — поиск, 7 — удаление, 8 — вставка, 9 — резерв, 0 — выход.

С помощью функций (1...4) просматривается программа по одной релейно-контактной цепочке. Предусмотрен поиск цепочек в программе (функция 6), содержащих заданный оператором символ.

Ввод новых уравнений осуществляется функцией 8. При ее работе оператор последовательно вводит строку комментария к данному уравнению, облегчающего чтение и позволяющего вносить необходимые пояснения. Далее оператору предлагается меню выходных команд (битовых, байтовых и арифметических), команд структурирования программ и других специальных инструкций. Если выбранная инструкция требует ввода дополнительных параметров (например, предуставки таймера), то система запрашивает их. Возможно параллельное включение нескольких выходных инструкций. Затем оператор вводит логическое уравнение: производится контроль синтаксиса, наличия символов в таблице имен, допустимости адреса с выдачей сообщений об ошибках (месте и виде ошибки). После окончания ввода на экране индицируется релейно-контактный эквивалент введенного уравнения.

СИМВОЛ	ОПЕРАНД	СИМВОЛ	ОПЕРАНД
S1	=10000	BP2.1	=10001
1KE3	=10007	SB10	=10012
KM3	=00024	Z26	=20410
V217	=00033	T2	=30416

Рис. 1. Пример составления таблицы имен

Для коррекции уравнений используется функция 5. Можно заменять и корректировать комментарии, выходные инструкции и уравнение с контролем ошибок. Функция 7 удаляет ненужную строку.

Для компенсации смещения адресов всей программы предусмотрена установка резерва (пустых инструкций [НОП]) после текущей строки (функция 9). Таких инструкций может быть от 0 до 999. Для выхода в главное меню используется функция 0.

Кроме логических уравнений программа ПК 32...ПК 248 должна иметь «маску обмена», содержащую служебную информацию. Для ее задания в главном меню существует специальный режим, при котором оператор проставляет необходимые для работы ПК параметры, показывает расстановку плат в каркасах ПК и имеет возможность предварительно записать необходимые данные в выбранные слова памяти ПК.

Режим «документирование» выдает на печать листинг программы из отдельных листов формата А3 (каждый лист имеет штамп с номером документа и автоматической нумерацией листов). Номер документа запрашивается при входе в режим. На первом листе печатается маска обмена и, если необходимо, текстовая информация. Второй лист содержит полную таблицу имен.

Лист программы (рис. 2) состоит из колонок: команд, полученных в результате трансляции уравнений; адресов операндов; адресов, по которым операнды будут записаны в память ПК; символических обозначений операндов и релейно-контактного эквивалента логического уравнения с необходимыми комментариями. Если уравнение или его релейно-контактный эквивалент не входит на данную страницу, они полностью переносятся на следующую. Уравнения в команды ПК транслируются непосредственно при печати (интерпретация).

Достоинство режима печати — возможность распечатки отдельных листов программы без смещения адресации остальных листов, что позволяет оперативно корректировать программу и получать измененный лист документации. При печати пустые инструкции [НОП], а также адреса не найденных в таблице имен символов печатаются как «...» для внесения изменений вручную. Режим допускает просмотр оттранслиро-

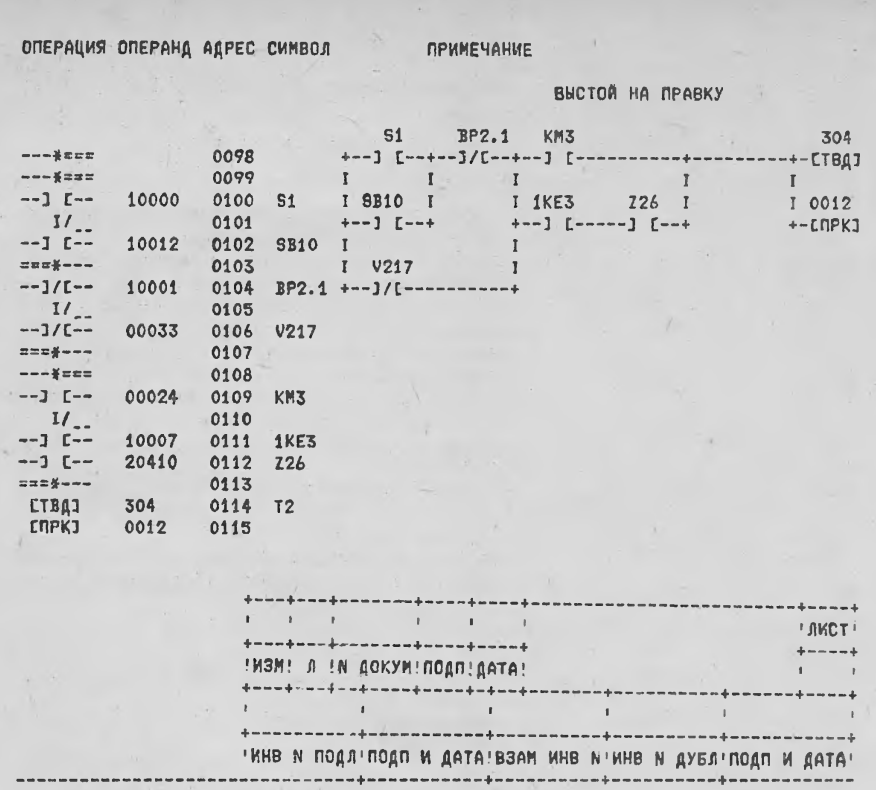


Рис. 2. Фрагмент выходного документа системы

рованной программы на дисплее (без печати).

Для хранения набранной программы используется режим «Запись-копирование» файла. Если при выходе в ОС редактируемый файл не был записан, система запрашивает оператора о необходимости сохранения файла.

Возможен дополнительный режим «Моделирование», позволяющий просматривать состояние выходных сигналов при принудительной установке входных сигналов в диалоговом режиме, т. е. имитировать работу ПК (сканирование программы с обработкой входов-выходов).

В настоящее время в систему для ПК 32...ПК 248 дополнительно введена непосредственная работа компьютера с контроллером по последовательному каналу. Этот режим сокращает время на ввод программы в ПК и исключает ошибки. Кроме того, возможна корректировка программы и данных непосредственно в ПК.

Система подготовки программ для ПК NS-915 отличается только формой выходного документа.

Телефон 273-00-31, доб. 689, Москва, Селезнев Валерий Петрович

Статья поступила 23.05.88

УДК 681.3.01

М. В. Руцков

МУЛЬТИПОРТОВАЯ ПАМЯТЬ ДЛЯ ПОСТРОЕНИЯ ВИДЕОКОМПЛЕКСОВ

Комплексирование — эффективный способ построения вычислительных систем для решения производственных и научно-исследовательских задач [1]. Объединение в комплекс готовых вычислительных средств (разнообразных ЭВМ, стандартных периферийных устройств, специализированных процессоров и контроллеров ввода — вывода) и дополнительно создаваемых устройств с требуемыми техническими характеристиками значительно снизит время и стоимость разработок.

Важное условие реализации комплексирования — совместимость вычислительных средств. Однако такая возможность существует лишь для ограниченного круга устройств, поэтому возникает необходимость в комплексировании разнородных устройств.

В качестве связующего ядра комплекса можно использовать мультипортовую память (МПП), выполненную в виде автономного устройства [2]. Доступ в память осуществляется с помощью адаптеров, выполняющих функции сопряжения интерфейсов. Связь через общее поле памяти дает возможность: гибко перестраивать комплекс (добавлять, исключать или заменять устройства), обеспечивать независимость периферийных устройств от интерфейсных характеристик конкретных ЭВМ, создавать многомашинные конфигурации и т. д.

При разработке реального устройства (видеокомплекса) ставилась задача сведения к минимуму трудоемкости его изготовления, что особенно важно при создании экспериментальных систем. Для указанных целей был выбран крейт КАМАК, который серийно выпускается промышленностью. Он имеет жесткую модульную конструкцию (25 модулей), оснащен мощным источником питания как для цифровых, так и для аналоговых микросхем и обеспечен набором макетных и ремонтных модулей. Более того, удалось полностью, без изменений, использовать межмодульные связи крейта.

Технические характеристики МПП

Число портов	16
Адресное пространство, Мбайт	2
Емкость ОЗУ, Мбайт	1
Пропускная способность, Мбайт/с	4
Формат обмена, байт	2
Быстродействие схемы приоритета, нс	125
Тип интерфейса	Синхронный

Структурная схема устройства приведена на рис. 1. Для подключения адаптеров выделены позиции 1...15, которые связаны с базовыми модулями единой магистралью для передачи адреса (21 бит), данных (16 бит), кода операции, специальных сигналов и двумя индивидуальными линиями — запроса и выборки.

Рассмотрим организацию модулей системы. В состав МПП входят четыре модуля ОЗУ общей емкостью 1 Мбайт. Формат обращения равен двум байтам. Память реализована на БИС ОЗУ К565РУ5Д динамического типа и имеет цикл обращения в 500 нс. Модули ОЗУ способны выполнять операции чтения и записи над словами (двумя байтами) и над байтами (в коде операции предусмотрен соответствующий признак).

Функция регенерации памяти реализована в модуле РПС. Каждые 8 мкс он запрашивает память в режиме чтения. Полный перебор 256 адресов строк во всех БИС ОЗУ происходит за 2 мс. Регенератор эквивалентен адаптеру АД-0.

Память состояний (256 слов) расположена в модуле РПС и выполнена на быстродействующих ИС ОЗУ К155РУ5. Ее основное назначение — хранение слов состояний, необходимых, в первую очередь, для синхронизации процедур обработки и обмена данными между абонентами (между устройствами, подключаемыми через адаптеры к МПП), а также для запоминания информации о прерываниях. В памяти состояний предусмотрено, кроме чтения и записи, выполнение операций установки и сброса битов. Для этого в коде операции выделено два соответствующих признака. Их сброс или установка происходит в позициях, которые отмечены Лог. 1 в слове, поступающем с шины данных. Память при этом работает в режиме чтение-модификация-запись с выполнением соответствующих логических операций.

Побитная обработка необходима, прежде всего, при обмене прерываниями, который осуществляется через специальную зону (память прерываний — ППР) в памяти состояний (рис. 2). Всем адаптерам в зоне выделено одно слово, биты которого предназначены для запоминания запросов прерывания от других адаптеров (номера адаптеров соответствуют номерам битов в слове). Обращение с любым кодом операции в область ППР приводит к логическому суммированию (после фазы модификации) всех разрядов адресуемого слова и выдаче полученного сигнала, отражающего наличие или отсутствие прерываний, на магистраль. Далее сигнал запоминается в соответствующем адресованному слову адаптере. Для этого в каждом адаптере предусмотрена своя логика декодирования адреса. Декодирование упрощается благодаря еще одному магистральному сигналу, указывающему на обращение в зону ППР. Таким образом, после получения сигнала прерывания адаптер может обратиться в ППР для определения его источника.

При отсутствии побитной обработки для установки или сброса бита прерывания требуется произвести следующую последовательность действий: чтение, модификацию бита а адаптере и запись результата. Однако в паузе между чтением и записью может начаться

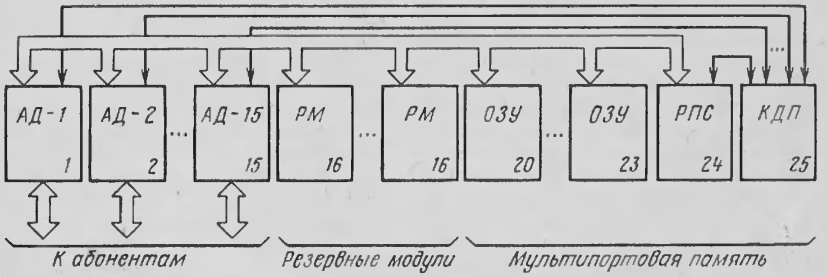


Рис. 1. Структура комплекса:
ОЗУ — оперативное запоминающее устройство;
РПС — регенератор и память состояний; КДП — контроллер доступа в память

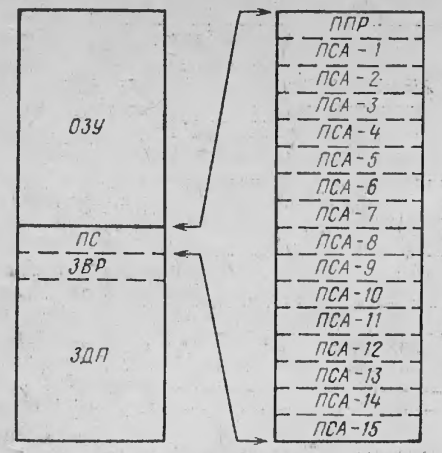


Рис. 2. Распределение адресного пространства

аналогичная процедура по тому же адресу со стороны другого адаптера. При этом бит, модифицированный первой процедурой, будет возвращен в исходное состояние после завершения второй, что может привести к потере запроса прерывания или к его «зависанию». Необходимо отметить, что прерывание, адресованное регенератору (АД-0), блокирует его работу.

Основные функции модуля КДП: выработка системных сигналов синхронизации с помощью задающего кварцованного генератора с частотой 8 МГц. С помощью пересчетных схем формируется набор тактовых импульсов с периодом 500 нс (время цикла модулей ОЗУ), имеющих разные фазы и скважности; все синхросигналы через буферные усилители подаются на магистраль и доступны в любом модуле; управление доступом в память. Каждый адаптер связан с модулем двумя индивидуальными линиями. По первой посылаются запросы на доступ в МПП, по второй возвращаются сигналы выборки, которые разрешают выбранному адаптеру произвести обмен с памятью. При работе системы могут возникать конфликтные ситуации: в модуль КДП одновременно поступает несколько запросов, на которые в каждом цикле необходимо вырабатывать только один ответный сигнал. Схема приоритета, входящая в состав модуля и имеющая 16 входов (по числу адаптеров), выполняет данные функции и допускает асинхронное поступление запросов. За 125 нс до завершения текущего цикла памяти фиксируются запросы на входном регистре схемы и к началу следующего вырабатывается сигнал выборки.

Эффективность работы системы зависит от алгоритмов обслуживания запросов. Предусмотрено несколько режимов функционирования схемы приоритета: скользящий, блочный и монопольный захват магистрали, фиксированный приоритет.

Для реализации указанных режимов схема приоритета содержит указатель номера старшей позиции. Адаптер с этим номером получает наивысший приоритет, который убывает с возрастанием номеров. После максимального номера (15) следует нулевой номер, образуется приоритетное кольцо. Указатель состоит из регистра, в котором хранится номер последнего обслуженного запроса, и сумматора.

В режиме скользящего приоритета содержимое регистра увеличивается на единицу (суммирование выполняется по модулю 16); последний обслуженный адаптер получает самый низкий приоритет и становится в конец очереди. Со старшей позиции осуществляется поиск ближайшего (наиболее старшего) активного (выставившего запрос) адаптера, номер которого после обслуживания загружается в регистр указателя. В следующем цикле процесс повторяется. Таким образом, по мере обслуживания запросов происходит скольжение указателя по приоритетному кольцу в сторону возрастания номеров. Данный режим обеспечивает одинаковые условия доступа всем абонентам системы.

В режиме блочного захвата магистрали работа сумматора блокируется соответствующим признаком в коде операции, который задается адаптером. Это позволяет данному адаптеру оставаться на самом высоком значении приоритета, в течение обмена блоком данных. Однако обмен должен выполняться без пауз, так как в отсутствие запроса значение наивысшего приоритета может перейти другому адаптеру. Например, 32-разрядному абоненту необходимо обмениваться данными с 16-разрядной МПП в два приема.

Для монопольного захвата магистрали адаптер (при получении доступа) выставляет сигнал, по которому блокируется работа сумматора и запрещается смена состояния регистра указателя. Адаптер закрепляет за собой наивысший приоритет и удерживает его в течение действия сигнала монопольного захвата. В отличие от предыдущего режима допускаются паузы в обмене, во время которых другие адаптеры могут работать с памятью.

Режим фиксированного приоритета аналогичен режиму монопольного захвата, но устанавливается оператором с панели модуля КДП.

Работа схемы приоритета связана также с анализом сигнала готовности, вырабатываемым каждым модулем, имеющим память (оперативную или постоянную) или регистры, расположенные в адресном пространстве системы. Модули ОЗУ и память состояний выдают сигнал готовности сразу при декодировании адреса, поскольку обращение к ним занимает один системный цикл. Более медленные устройства выставляют сигнал с задержкой, в течение которой схема приоритета блокирует свою работу. Одновременно запускается таймер для измерения времени остановки. Если сигнал готовности приходит слишком поздно или совсем не вырабатывается при нарушении связей или неправильной адресации (в незанятую область адресного пространства), то таймер формирует, отсчитав заранее заданное время, сигнал ошибки доступа. Этот сигнал выдается на магистраль, фиксируется и выводится на индикацию в адаптере-инициаторе данного обмена. Далее происходит деблокирование схемы приоритета и доступ в память в принудительном порядке (путем сброса в ноль указателя) предоставляется АД-0 (регенератору). Такая мера необходима для защиты памяти от потери данных, которая может произойти при сбоях регенерации.

В системе предусмотрен еще один механизм поддержки и контроля регенерации. Регенератор каждые 8 мкс требует доступ в память. При высокой интенсивности обмена абонентов с МПП обслуживание этих требований может задерживаться (более чем на 8 мкс). В этом случае начинается накопление (подсчет) нереализованных запросов с одновременным переходом регенератора в режим непрерывных требований доступа (в каждом цикле). Если при этом число нереализованных запросов продолжает расти и достигает первого порога, то включается механизм блочного захвата магистрали. В случае превышения второго порога вырабатывается сигнал «сбой регенерации», который запоминается и индицируется модулем РПС. С помощью перемычек можно задать оба порога.

Адресное пространство системы занимает 2 Мбайта (21 бит адреса).

Под ОЗУ отводится 1 Мбайт. В зоне дополнительной памяти (1 Мбайт) первые 512 байт занимает память состояний (ПС), которая разбита на 16 зон по 16 слов. Каждому адаптеру выделена своя зона — память состояния адаптера (ПСА1...ПСА15). Регенератору такая память не нужна, поэтому в его зоне расположена память прерываний (ППР). Далее следует зона внешних регистров (ЗВР 512 байт). При обращении в эту зону модуль РПС вырабатывает соответствующий сигнал, который по магистральной линии поступает во все модули системы. Это упрощает логику декодирования адреса внешних регистров.

Оставшееся адресное пространство предоставляется пользователям для подключения дополнительной оперативной или постоянной памяти. Для этих целей в составе системы имеются четыре позиции резервных модулей (РМ).

Наиболее важный вопрос при создании комплексов — выбор способа сопряжения вычислительных средств с мультипортовой памятью. В этом плане предпочтение отдано способам, использующим прямую адресацию памяти с применением механизмов расширения. Один из возможных вариантов такого механизма реализован в устройстве,

разработанном для сопряжения системы с микроЭВМ «Электроника 60». В адресном пространстве машины выделена специальная зона связи (8 Кбайт, 13 бит адреса), при обращении в которую происходит вызов МПП. Девять младших битов без изменений транслируются на адресную магистраль системы. Старшие 4 бит выбирают один из 16 регистров переадресации (регистры выполнены на ИС ОЗУ К155РУ2), размер которых равен 14 бит (12 бит адреса и два признака побитной обработки — установка и сброс). Содержимое регистров поступает на адресную магистраль и образует полный адрес с ранее выведенными младшими битами. Предварительно регистр загружается процессором ЭВМ. Таким образом, процессор в режиме прямой адресации может обращаться к любым 16 сегментам МПП (размер сегмента составляет 512 байт).

Аналогичным образом реализовано сопряжение с микроЭВМ на базе микропроцессора К1810ВМ86, большое адресное пространство которого (1 Мбайт) позволило выделить окно размером 256 Кбайт. Оно разбито на четыре сегмента по 64 Кбайт каждый. Суммирование регистра переадресации с исполнительным адресом (индексирование) позволило настраивать сегменты с шагом в 32 слова. Для реализации связи с МПП при побайтовых операциях предусмотрена коммутация байта данных с младших линий на старшие при нечетных адресах.

Устройства сопряжения выполняются в виде двух модулей — буфера и адаптера, соединенных между собой кабелем (1—3 м). Конструктивно буфер располагается в подключаемой ЭВМ, а адаптер — в МПП.

Использование механизмов расширения адресации не вызывает затруднений при работе с ОС микроЭВМ, так как при генерации окна связи не учитываются, а обращение к МПП осуществляется на уровне подпрограмм, написанных на языке ассемблера и включенных в библиотеки языков высокого уровня.

Однако способы адресации, реализованные в стандартных микроЭВМ, и рассмотренные механизмы расширения предназначены, в основном, для работы с линейными массивами и малоэффективны при обработке изображений.

Практический опыт создания ПО для видеоанализа и машинной графики показывает, что основное время процессора затрачивается на формирование адресов элементов изображения и преобразование формата видеоданных [3]. Перенесение этих функций на аппаратный уровень позволяет повысить скорость обработки без потери универсальности микроЭВМ на один, два порядка.

Для этого разработано **устройство сопряжения «Пиксел»**, которое предоставляет пользователю возможность

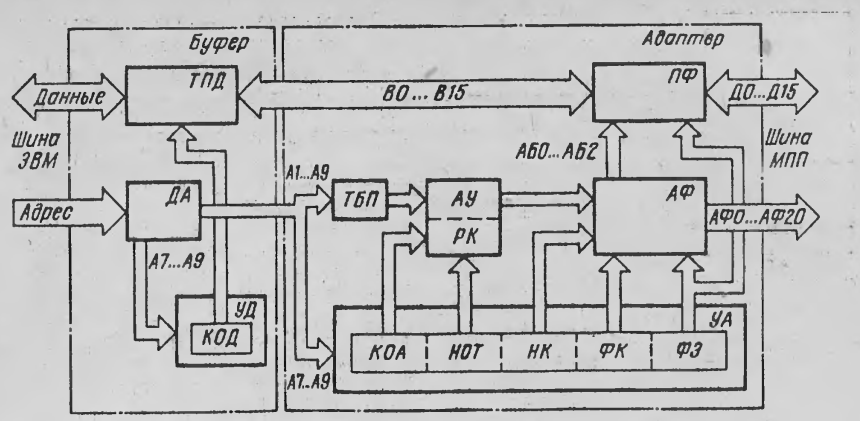


Рис. 3. Блок-схема устройства «Пиксел»

двухкоординатной адресации данных в пределах выбранного видеокадра (рис. 3). При этом заранее определяются параметры кадра (размер, местоположение в памяти и разрядность элементов изображения).

Косвенный способ адресации по текущим значениям опорных координат (X, Y) — основа доступа к видеоданным. Он сочетается с возможностью задания двухкоординатного смещения или автоприращения (ΔX, ΔY), что позволяет реализовать режим адресации окрестности опорной точки.

Отличительная особенность предложенного адресного механизма — поэлементное обращение к матрице изображения, хотя видеоданные хранятся в памяти в упакованном виде (несколько элементов в слове).

Кроме операций над адресами в устройстве реализованы форматно-редактирующие операции и табличные преобразования данных (при разрядности данных не более 8 бит).

Буфер устройства реализует функции связи с шиной микроЭВМ и табличного преобразователя данных; адаптер фактически представляет собой адресный спецпроцессор, выполняющий функции формирования исполнительных адресов «на проходе» при обращении микроЭВМ к МПП.

Для доступа к МПП в адресном пространстве микроЭВМ выделено окно размером 512 слов (по 2 байт), разбитое на восемь зон связи (по 64 слова). При обращении в указанную область срабатывает дешифратор адреса (ДА), который транслирует девять младших битов (А1...А9) в адаптер. Этот адрес используется для выбора из таблицы приращений (ТБП) значений ΔX и ΔY, которые представлены 12-разрядными двоичными числами (старший бит знаковый). Таблица реализована на ИС ОЗУ К132РУ8.

Старшие биты адреса (А7...А9) определяют номер зоны связи. Они поступают в таблицу управления адресом (УА) для выбора соответствующего слова, которое состоит из следующих полей:

КОА — код операции над адресами (3 бит) определяет режимы адресации;

НОТ — номер опорной точки (3 бит) выбирает одну из восьми пар опорных координат X и Y, расположенных в регистровом блоке координат (РК);

НК — номер кадра (8 бит) указывает местоположение видеокадра в МПП. Кадры должны размещаться в памяти кратно своей информационной емкости;

ФК — формат кадра (2 бит) задает один из четырех размеров кадра: 256×256, 512×512, 1024×1024, 2048×2048 элементов;

ФЭ — формат элемента (3 бит) определяет разрядность элемента изображения: 1, 2, 4, 8, 16 бит.

Значения выбранных опорных координат и приращений поступают в 12-разрядное арифметическое устройство (АУ), которое предназначено для вычисления исполнительных координат. Регистровый блок координат и АУ реализованы на микропроцессорных секциях К1804ВС1, их работой управляют биты-признаки поля КОА: первый задает режимы смещения (опорные координаты не изменяются) или автоприращения (опорные координаты модифицируются), второй определяет порядок выдачи исполнительных координат — до или после фазы суммирования; третий одинарной или двойной точности.

Опорные координаты и приращения дополнительно содержат 12-разрядные дробные части, занимая в РК и ТБП соответственно по две смежные ячейки.

Из АУ исполнительные координаты поступают в адресный формирователь (АФ), где вместе с номером кадра (К0...К7) преобразуются в 24-разрядный адрес элемента. Такая разрядность определяется минимальным форматом элемента изображения (1 бит) и максимальным размером адресного пространства МПП (2 Мбайт=16 Мбит). Необходимо отметить, что значения адреса элемента всегда кратны его формату, т. е. 1-разрядные элементы следуют подряд, 2-разрядные — по четным адресам и т. д.

Видеоданные расположены в памяти стандартным образом (в упакованном

Таблица 1

Преобразование адреса

Параметры кадра	АФ																				АБ		
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	2	1	0
256×256×1	К7	К6	К5	К4	К3	К2	К1	К0	У7	У6	У5	У4	У3	У2	У1	У0	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0
256×256×2	К7	К6	К5	К4	К3	К2	К1	У7	У6	У5	У4	У3	У2	У1	У0	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0
256×256×4	К7	К6	К5	К4	К3	К2	У7	У6	У5	У4	У3	У2	У1	У0	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0
256×256×8	К7	К6	К5	К4	К3	У7	У6	У5	У4	У3	У2	У1	У0	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0	0
256×256×16	К7	К6	К5	К4	У7	У6	У5	У4	У3	У2	У1	У0	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0	0	0
512×512×1	К7	К6	К5	К4	К3	К2	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0
512×512×2	К7	К6	К5	К4	К3	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0
512×512×4	К7	К6	К5	К4	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0
512×512×8	К7	К6	К5	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0	0
512×512×16	К7	К6	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0	0	0
1024×1024×1	К7	К6	К5	К4	У9	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х9	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0
1024×1024×2	К7	К6	К5	У9	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х9	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0
1024×1024×4	К7	К6	У9	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х9	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0
1024×1024×8	К7	У9	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х9	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0	0	0
2048×2048×1	К7	К6	У10	У9	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х10	Х9	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0
2048×2048×2	К7	У10	У9	У8	У7	У6	У5	У4	У3	У2	У1	У0	Х10	Х9	Х8	Х7	Х6	Х5	Х4	Х3	Х2	Х1	Х0	0

виде, по последовательным адресам, в порядке возрастания номеров: разрядов в элементе, элементов в строке и строк в кадре), а параметры видеокадра задаются только значениями из степенного ряда числа два, поэтому адресные преобразования выполняются с помощью операций сдвига и маскирования (табл. 1).

Из 24 разрядов сформированного адреса только старшие (АФ0...АФ20) используются для обращения в память, имеющую байтовую организацию. Младшие три разряда (АБ0...АБ2) определяют адрес элемента внутри байта.

Время всех адресных операций, выполняемых в узлах АУ и АФ, не превышает 0,5 и 1 мкс для режимов с одинарной и двойной точностью соответственно. Параллельно осуществляется контроль текущих значений исполнительных координат. Если они выходят за пределы назначенного кадра, то обмен с МПП блокируется.

Реализация поэлементного способа доступа в память, кроме адресных операций, требует преобразования формы представления видеоданных, которое выполняется в узле преобразования формата (ПФ). Это связано с тем, что микроЭВМ работает с индивидуальными элементами изображения, каждый из которых независимо от своего формата занимает целое машинное слово (2 байт); а номера разрядов элемента совпадают с номерами разрядов в слове. Неиспользуемые старшие разряды (при формате элемента менее 16 бит) заполняются нулями.

Узел ПФ с одной стороны связан с МПП шиной данных (Д0...Д15), а с другой — внутренней шиной (В0... В15) с буфером. Элементы изображения форматом 16 бит проходят через ПФ без изменений. В остальных случаях выполняется преобразование формата (коды управления ФЭ и АБ).

В режиме ввода (табл. 2) выполняется операция чтения байта данных из МПП с последующей коммутацией выбранного элемента на младшие позиции внутренней шины.

Процедура вывода (табл. 3) требует двух циклов обращения к памяти. Сначала выполняется чтение байта данных, затем — его модификация и запись на исходное место в МПП. При этом модификация осуществляется с помощью «вставки» выводимого элемента в адресуемую позицию внутри байта.

Кроме специализированных способов адресации в устройстве предусмотрено табличное преобразование данных

Таблица 2

Преобразование формата данных при вводе

Формат элемента, бит	АБ			В7	В6	В5	В4	В3	В2	В1	В0
	2	1	0								
1	0	0	0	0	0	0	0	0	0	0	Д0
1	0	0	1	0	0	0	0	0	0	0	Д1
1	0	1	0	0	0	0	0	0	0	0	Д2
1	0	1	1	0	0	0	0	0	0	0	Д3
1	1	0	0	0	0	0	0	0	0	0	Д4
1	1	0	1	0	0	0	0	0	0	0	Д5
1	1	1	0	0	0	0	0	0	0	0	Д6
1	1	1	1	0	0	0	0	0	0	0	Д7
2	0	0	0	0	0	0	0	0	0	Д1	Д0
2	0	1	0	0	0	0	0	0	0	Д3	Д2
2	1	0	0	0	0	0	0	0	0	Д5	Д4
2	1	1	0	0	0	0	0	0	0	Д7	Д6
4	0	0	0	0	0	0	0	Д3	Д2	Д1	Д0
4	1	0	0	0	0	0	0	Д7	Д6	Д5	Д4
8	0	0	0	Д7	Д6	Д5	Д4	Д3	Д2	Д1	Д0

Таблица 3

Преобразование формата данных при выводе

Формат элемента, бит	АБ			Д7	Д6	Д5	Д4	Д3	Д2	Д1	Д0
	2	1	0								
1	0	0	0	Д7	Д6	Д5	Д4	Д3	Д2	Д1	В0
1	0	0	1	Д7	Д6	Д5	Д4	Д3	Д2	В0	Д0
1	0	1	0	Д7	Д6	Д5	Д4	Д3	В0	Д1	Д0
1	0	1	1	Д7	Д6	Д5	Д4	В0	Д2	Д1	Д0
1	1	0	0	Д7	Д6	Д5	В0	Д3	Д2	Д1	Д0
1	1	0	1	Д7	Д6	В0	Д4	Д3	Д2	Д1	Д0
1	1	1	0	Д7	В0	Д5	Д4	Д3	Д2	Д1	Д0
1	1	1	1	В0	Д6	Д5	Д4	Д3	Д2	Д1	Д0
2	0	0	0	Д7	Д6	Д5	Д4	Д3	Д2	В1	В0
2	0	1	0	Д7	Д6	Д5	Д4	В1	В0	Д1	Д0
2	1	0	0	Д7	Д6	В1	В0	Д3	Д2	Д1	Д0
2	1	1	0	В1	В0	Д5	Д4	Д3	Д2	Д1	Д0
4	0	0	0	Д7	Д6	Д5	Д4	В3	В2	В1	В0
4	1	0	0	В3	В2	В1	В0	Д3	Д2	Д1	Д0
8	0	0	0	В7	В6	В5	В4	В3	В2	В1	В0

(ТПД), выполняемое в буфере. Максимальный формат преобразуемых элементов изображения — 8 бит. В ТПФ расположено 8 таблиц размером 256×8 бит, реализованных на ИС ОЗУ К132РУ8.

Параметры преобразования загружаются в таблицу управления данными (УД), в которой каждой зоне связи выделено по два слова — для режимов ввода и вывода. Управляющее слово содержит поле КОД (код операции над данными, 4 бит), состоящее из номера преобразовательной таблицы (3 бит) и признака ее включения. При пассивном значении признака данные проходят через узел ТПД без изменений.

Таким образом, устройство «Пиксел» предоставляет пользователю гибкие возможности для эффективной обработки изображений.

Способ двухкоординатного смещения задает любую форму апертуры, число точек в которой при использовании одной зоны связи равно 64. Это число можно увеличить, если задействовать несколько зон связи с указанием в них одинакового номера опорной точки.

Способ двухкоординатного автоприращения наиболее эффективен для реализации спрайтовой графики. В сочетании с режимом вычисления адресов с двойной точностью он позволяет выполнять следующие процедуры: вычерчивание линий (с произвольным наклоном), закрашивание объектов, прокрутку, масштабирование и повороты изображений.

Разбиение окна доступа на зоны связи обеспечивает: использование нескольких опорных точек в пределах одного видеокадра или работу с многослойными изображениями (одна опорная точка для всех слоев). Элементы каждого слоя могут быть произвольных форматов.

Табличные преобразования данных дают возможность реализации любых одноопернадных операций (логарифмирование, извлечение корня, арифметические действия с константами). Эффективно решается задача нелинейных преобразований, а также согласования (сжатия, расширения и перекодировки) форматов элементов изображения. Аппаратная реализация указанных функций сокращает объем ПО и резко повышает скорость обработки видеоданных. Например, микроЭВМ «Электроника 60» может выполнить поворот изображения 512×512 элементов на произвольный угол с масштабированием за 2,5 с.

Дальнейшее повышение производительности и расширение функциональных возможностей видеокомплексов достигается за счет подключения к МПП специализированных видеопроцессоров и контроллеров ввода-вывода изображений. Разработаны следующие устройства:

«Телекадр» — для ввода-вывода изображений в формате 256×256×4 бит (время ввода-вывода 20 мс);

«Фильтр» — для нелинейной обработки полутоновых изображений локальным оператором на окне размерами 3×3 элемента (время обработки кадра 100 мс);

«Контур» — для составления контурного описания по изображению и обратной процедуры (быстродействие — 0,5 мкс на одну точку контура);

«Апертура» — для обработки бинарных изображений с помощью локальных линейных операторов произвольного вида (быстродействие — 0,25 мкс на одну точку апертуры);

«Спрайт» — для вывода изображений (формат 400×300×4 бит, 16 произвольных цветов) с аппаратной реализацией режима спрайтовой графики.

Модули ввода-вывода и обработки видеосигналов работают в режимах квадратного и гексогонального представлений элементов изображения.

Телефон 412-94-22,
Москва (после 19 ч.
Руцков Михаил Вадимович)

ЛИТЕРАТУРА

1. Байцер Б. Архитектура вычислительных комплексов.— М.: Мир, 1974.
2. Руцков М. В. Об одном подходе к проектированию систем технического зрения // Сб. Проблемы системного исследования и построения гибких автоматизированных производств.— М.: ВНИИСИ, 1986.
3. Бутаков Е. А., Островский В. И., Фадеев И. Л. Обработка изображений на ЭВМ.— М.: Радио и связь, 1987.

Статья поступила 19.05.88

УДК 681.772.7.049.771.14

В. И. Матвеенко, Ю. Г. Староверов

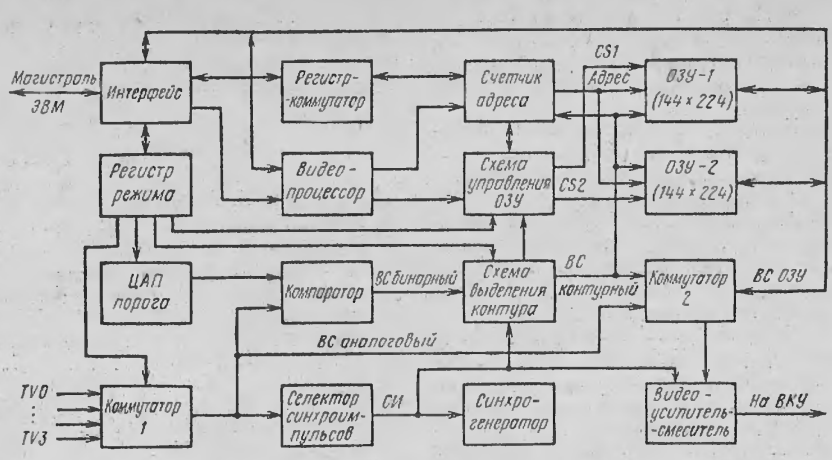
БИНАРНАЯ СИСТЕМА ТЕХНИЧЕСКОГО ЗРЕНИЯ ДЛЯ ПРОМЫШЛЕННОГО ПРИМЕНЕНИЯ

Решение большого числа производственных задач, связанных с распознаванием объектов, измерением геометрических характеристик, визуальным контролем изделий, технологических процессов и т. д., может быть возложено на системы технического зрения (СТЗ). Такие системы для промышленного применения должны обладать высоким быстродействием (не $<$ 1 с), при обеспечении требуемых точностных характеристик, и высокой надежностью [1]. В настоящее время наиболее развиты бинарные СТЗ (обрабатывающие две градации яркости изображения), обладающие большим быстродействием и простотой аппаратных реализаций.

Аппаратная часть рассматриваемой СТЗ включает стандартные телекамеры, серийную микроЭВМ типа «Электроника 60», устройство ввода видеоинформации (УВВИ) и видеоконтрольное устройство (ВКУ). УВВИ — единственный нестандартный блок системы — выполнено в виде единого встраиваемого в микроЭВМ модуля, что увеличивает надежность работы и улучшает ремонтопригодность СТЗ в целом. УВВИ обеспечивает подключение одновременно до четырех телекамер (на основе видиконов или ПЗС-матриц) с обработкой в режиме разделения времени. Основные функции предобработки изображения решаются в УВВИ

Назначение адресов

Адрес	Регистр	Разряд	Выполняемые функции
177340	Режим	00—07	Порог квантования
		08	Выбор ввода четных/нечетных строк изображения
		09—10	Выбор ОЗУ-1 или ОЗУ-2
		11	Готовность УВВИ
		12—13	Выбор телекамеры (TV0...TV3)
		14	Выбор фона (светлый/темный)
		15	Управление схемой выделения контура
177342	Адреса	00—15	Адрес точки изображения в ОЗУ
177344	Данных 1	00	Запись/чтение видеоинформации в ОЗУ-1
177346	Данных 2	00	Запись/чтение видеоинформации в ОЗУ-2
177350	Загрузки	00	Загрузка полукадра изображения или обнуление ОЗУ
177352	Видеопроцессора	00—01	Управление видеопроцессором (поиск изображения, анализ связности)
177354	ВКУ	00	Управление выводом изображения из ОЗУ на ВКУ
177356	Площади	00	Подсчет площади или периметра изображения



Структурная схема устройства ввода видеоинформации

аппаратными средствами, что позволяет обрабатывать сложные визуальные сцены за время не более 1 с при дискретности разложения изображения 144×224 точек или 288×224.

Отличительные особенности УВВИ: полная синхронность ввода изображения, даже в случае работы телекамеры от внутреннего автономного генератора; наличие схемы оконтуривания, работающей в темпе телевизионной развертки и внутреннего видеопроцессора, обеспечивающего быстрый поиск изображения на растре и проведение анализа связности контурной линии.

Для управления УВВИ и обмена информацией с центральным процессором (ЦП) в адресном пространстве микроЭВМ отведено восемь адресов (см. таблицу). Инициализация устройства ввода на конкретные условия работы осуществляется ЦП через регистр режима. В частности, восемь младших разрядов регистра задают порог квантования видеосигнала (ВС) по уровню. Разряд 8 определяет ввод изображения из-под четных или нечетных строк полукадра, что необходимо при работе с ПЗС-телекамерами на основе среднеформатных матриц, где в половине строк каждого полукадра отсутствует видеоинформация. Разряды 9 и 10 определяют, в какое из двух буферных ОЗУ в данном режиме загружается видеоинформация либо из какого ОЗУ изображение выводится на ВКУ. Разряд 11 регистра режима доступен ЦП только по чтению и определяет готовность УВВИ к обмену информацией с микроЭВМ. Этот разряд сбрасывается при загрузке полукадра изображения, обнулении ОЗУ, а также в течение четных полукадров при выводе изображения из ОЗУ на ВКУ. Разряды 12 и 13 управляют подключением к УВВИ одной из четырех телекамер (TV0...TV3); разряд 14 — инверсией бинарного видеосигнала на выходе компаратора, что необходимо при обработке изображений на светлом или темном фоне. Последний 15-й разряд регистра режима позволяет при необходимости заблокировать схему выделения контура, при этом в ОЗУ видеоинформации будет загружаться полное бинарное изображение.

Синхронизация работы всего УВВИ осуществляется от синхрогенератора (см. рисунок). Как правило, в аналогичных УВВИ синхрогенератор стабилизируется кварцевым резонатором, что дает высокую временную стабильность. Однако стандартные телекамеры работают от внутреннего автономного генератора, вследствие чего фаза синхросигнала устройства ввода по отношению к фазе видеосигнала телекамеры является случайной функцией. Это приводит к тому, что каждая строка видеоинформации вводится в ЭВМ с точностью ± 1 дискрет. В УВВИ автоподстройка фазы синхрогенератора осуществляется с приходом каждого строчного синхроимпульса (СИ), выделяемого селектором из полного ВС. Это позволило исключить указанную погрешность ввода изображения и аппаратно реализовать схему выделения контура.

Схема выделения контура — разновидность анизотропного фильтра с апертурой типа «крест» [2]. Алгоритм работы схемы: центральная точка обнуляется в том случае, если все четыре соседние точки равны 0 или 1. Такой алгоритм дает возможность не только выделять контур, но и фильтровать отдельные шумовые точки.

В *ОЗУ видеоинформации* хранится один полукадр изображения, загружаемый в темпе телевизионной развертки. Наличие двух одинаковых ОЗУ видеоинформации предоставляет дополнительные возможности. В частности, одно из ОЗУ может хранить текущее изображение, а второе — эталонное либо изображение, обработанное микроЭВМ, что визуализирует и облегчает процедуру отладки алгоритмов. В более сложных применениях дополнительное ОЗУ при наличии соответствующих средств программной поддержки обеспечивает использование ВКУ как графического монитора. Оба ОЗУ имеют адресную организацию, моделирующую структуру растра. Поскольку ОЗУ видеоинформации располагается не в адресном пространстве микроЭВМ, доступ ЦП к видеоинформации осуществляется за два цикла обращения. В первом цикле ЦП в регистр адреса записывает адрес требуемой точки изображения, который запоминается в регистре-коммутаторе и одновременно в счетчике адреса (см. рисунок). Для удобства написания программ поддержки УВВИ схема управления ОЗУ содержит дешифратор адреса, преобразующий младший байт адреса в координату столбца (0...223), а старший байт — в координату строки (0...143) требуемой точки изображения [4]. Во втором цикле обращения ЦП через регистр данных считывает или записывает значение одной точки изображения. Поскольку такая процедура обмена в некоторых случаях требует значительных временных затрат, основные функции предобработки изображения возложены на видеопроцессор (ВП).

Видеопроцессор работает в двух режимах: поиск и анализ связности. В режиме поиска изображения ЦП заносит в счетчики стартовый адрес поиска и ВП последовательно опрашивает точки вдоль каждой строки с периодом 220 нс. Адрес первой найденной ненулевой точки запоминается в регистре-коммутаторе и считывается ЦП. Во втором режиме ВП анализирует связность контурного изображения, т. е., двигаясь вдоль контурной линии, выдает в микроЭВМ адреса точек, расположенных на контурной линии друг за другом [3]. Необходимость в проведении анализа связности возникает, например, в тех случаях, когда для анализа изображения используются структурные признаки, основанные на характере изменения контурной линии. Это позволяет проводить обработку сложных визуальных сцен, например в случае перекрывающихся объектов. Цикл работы ВП включает в себя опрос восьми точек, соседних с текущей точкой, выбор по определенному алгоритму точки перехода и занесение ее адреса в регистр-коммутатор. Обращаясь к регистру-коммутатору, ЦП считывает адрес найденной точки перехода. При обнаружении на контуре точки ветвления, достижении границы растра или конца контурной линии ВП выдает специальные коды. Цикл работы ВП во втором режиме составляет 3,5 мкс, что примерно на два порядка производительнее программного способа предобработки.

Некоторые задачи, решаемые СТЗ, требуют быстрого подсчета площади и периметра изображения, используемых в качестве простейших признаков распознавания, при построении гистограмм для выбора оптимального порога компарирования и т. д. Как правило, устройства ввода содержат для этой цели дополнительные счетчики. Для минимизации аппаратурных затрат в данном УВВИ функция подсчета площади и

периметра возложена на счетчик адреса. При занесении кода в регистр подсчета площади счетчики строк и столбцов, входящие в счетчик адреса, объединяются в один 16-разрядный счетчик, на вход которого подается ВС со схемы выделения контура. При заблокированной схеме выделения контура подсчитывается площадь, при разблокированной — периметр. Подсчет проводится в течение одного полукадра (20 мс), по окончании которого подсчитанное значение заносится в регистр-коммутатор и считывается ЦП.

Для удобства обслуживания системы видеоинформация может быть выведена на ВКУ в следующих видах: исходное изображение с телекамеры, полное бинарное, контурное, а также изображение из любого из двух ОЗУ видеоинформации.

Базовое программное обеспечение СТЗ включает программы тестирования УВВИ и программы нижнего уровня, непосредственно поддерживающие аппаратную часть системы. Программы верхнего уровня разрабатываются для конкретного применения. В частности, разработано программное обеспечение для поиска начала сварного шва.

Устройство ввода видеоинформации выполнено в объеме одной стандартной платы конструктива «Электроника 60» (240 × 280 мм) и содержит в общей сложности 103 микросхемы.

Система технического зрения может применяться для робототехники, распознавания объектов, измерения, контроля. Точность определения линейных координат объекта в зависимости от условий измерения составляет 0,2... 0,5 % от ширины поля зрения, среднее время обработки изображения — 500 мс.

Телефон 277-28-39, Москва

ЛИТЕРАТУРА

1. Техническое зрение роботов / Под ред. А. Пью. — М.: Машиностроение, 1987.
2. СБИС для распознавания образов и отработки изображений / Под ред. К. ФУ. — М.: Мир, 1988.
3. Заявка № 59—22994 (Япония), 1984.
4. Матвеенко В. И., Староверов Ю. Г. Полутоновая система технического зрения // Микропроцессорные средства и системы. — 1987. — № 2. — С. 68—69.

Статья поступила 15.03.88

УДК 681.327.8.

Ю. П. Юрченко, А. П. Казмирук

СОПРЯЖЕНИЕ СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ СТЗ-1 С МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Система технического зрения СТЗ-1 обеспечивает ввод в микроЭВМ «Электроника 60» информации о яркости и координатах исследуемой точки изображения. Она входит в состав телевизионного комплекса, выпускаемого промышленностью для автоматизации визуального технологического контроля*. МикроЭВМ подключается к системе через два устройства параллельного обмена И215КС-180-032.

В данной статье предлагается схема подключения СТЗ-1 к микроЭВМ «Электроника 60» с помощью одного устройства параллельного обмена.

Для этого необходимо модифицировать только схему жгута (см. рисунок). Входной регистр устройства параллельного обмена используется для приема кода яркости Q1...Q6, а 16 линий выходного регистра — для установки адреса контролируемого элемента. При этом восемь младших разрядов задействованы под шину Вход X2... Вход X9, а остальные восемь разрядов — под шину Вход У2... Вход У9. Опыт работы с установкой показал, что видеоинформация считывается в строке не с 512 точек, а только с 360 (разряд У10 при установке кода У не используется). Поэтому контакты выходного разъема установки Вход У10 и Общий нужно замкнуть.

Соединение контактов Общий и Вход У1 уменьшит в два раза число опрашиваемых микроЭВМ элементов строки изображения. Суммарное число контролируемых элементов в строке станет равным 180. При аналогичном подключении контактов Вход X1 и Вход X10 информация об изображении будет считываться только во время одного из полукадров и только по 510 строку кадра включительно. Таким образом, данная схема разводки жгута приведет к потерям нужной информации.

Этих потерь можно избежать, если для установки кода X использовать линии разрядов РС 00 и РС 01 регистра состояния. Тогда восемь младших разрядов кода X можно установить с помощью линий выходного регистра, а X9, X10 — линий разрядов РС 00, РС 01 регистра состояния устройства параллельного обмена. Размер формируемой матрицы изображения будет равен 180×574 мм.

Телефон 441-91-04, Киев

Статья поступила 14.06.88

* Система технического зрения СТЗ-1. Техническое описание и инструкция по эксплуатации. ЦБМ 1.456.000ТО, 1985.

XP4

Цепь	Конт.	
Требов. А	16	1
Вход X1	18	2
Вход X2	22	3
Вход X3	32	4
Вход X4	33	5
Вход X5	34	6
Вход X6	35	7
Вход X7	36	8
Вход X8	37	9
Вход X9	38	10
Вход X10	41	11
Корпус	39,40	12
Р1	26	13
Р2	27	14
Р3	28	15
Р4	29	16
Р5	30	17
Р6	31	18

XP5

Цепь	Конт.	
Запись	7	19
Вход У2	10	20
Вход У3	11	21
Вход У4	12	22
Вход У5	13	23
Вход У6	14	24
Вход У7	15	25
Вход У8	16	26
Вход У9	17	27
Ввод данных	6	28
Вход У1		12
Вход У10		12

XS

	Конт.	Цепь
1	8	Требов. А
2	49	ВД00
3	51	ВД01
4	29	ВД02
5	33	ВД03
6	2	ВД04
7	1	ВД05
8	14	ВД06
9	10	ВД07
10	53	РС00
11	60	РС01
12	18,20 22,24	Общий
13	47	ВВ00
14	45	ВВ01
15	27	ВВ02
16	31	ВВ03
17	4	ВВ04
18	6	ВВ05
19	21	Требов. В
20	59	ВД08
21	46	ВД09
22	52	ВД10
23	19	ВД11
24	43	ВД12
25	54	ВД13
26	35	ВД14
27	48	ВД15
28	15	Ввод данных

Электрическая схема жгута

УДК 681.3.06

Д. С. Коврига, С. М. Гладченко

РАСШИРЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОМЫШЛЕННОГО РОБОТА «ЭЛЕКТРОНИКА НЦТМ-01»

Существует несколько версий программного обеспечения промышленного робота «Электроника НЦТМ-01» для робототехнологических комплексов (РТК) на базе станков с числовым программным управлением (ЧПУ).

Предлагаемая версия ПО предназначена для создания РТК на базе токарных станков с ЧПУ типа Н22. Ее функциональные подпрограммы обеспечивают:

ручное управление манипулятором в режиме наладки с помощью команд, подаваемых с клавиатуры дисплея 15ИЭ-00-013;

Таблица 1

Таблица соответствия символов клавиатуры выполняемым РТК командам

Символ клавиатуры	Команда
	Ручного управления
→	Движение механизма А вправо
←	Движение механизма А влево
↓	Движение механизма С от станка
↑	Движение механизма С к станку
↥	Движение механизма Д вверх
↧	Движение механизма Д вниз
⌐	Поворот механизма В к патрону станка
⌟	Поворот механизма В к кассете
↖	Зажим схвата детали
↓	Разжим схвата детали
←┤	Зажим схвата заготовки
├→	Разжим схвата заготовки
АР2 С1	Поворот схватов
З	Зажим патрона станка
Р	Разжим патрона станка
Т	Конец режима ручного управления
М	Вызов режима ручного управления

```
.TITLE ROB1
.ENABLE LC,GBL
;*****************************************************************
;*Программа ручного управления манипулятором "ЭЛЕКТРОНИКА НЦТМ-01"*
;*Разработчик-КОВРИГА Д.С.                                      *
;*Адрес:330118,г.Запорожье,проспект 40-летия ПОБЕДЫ 17,кв.44     *
;*****************************************************************
;
; Вывод на экран подсказки
;
ROB1::   MOV     #TEXT,R0          ; адрес текста в R0
M0:      TSTB    @#177564          ; готов дисплей ?
         BPL     M0                ; нет--->M0
         MOVB    (R0)+,@#177566    ; вывод символа на экран
         BNE     M0                ; конец текста ? нет--->M0
;
; Ввод данных с клавиатуры
;
VVOD:    MOV     #7777,R1          ; R1-счётчик цикла ожидания готовности
         CLR     R0                ; R0-код символа
VV1:     TSTB    @#177560          ; нажата клавиша ?
         BMI     VV2               ; да--->VV2
         SOB     R1,VV1            ; нет--->ждать нажатия
;
; Если клавишу отпустили
;
STOP:    CLR     @#167772          ;-стоп манипулятора
         CLR     @#167762          ;/
         BR      VVOD              ; на ожидание нажатия--->VVOD
VV2:     MOVB    @#177562,R0       ; ввод кода клавиши в R0
         BIC     #177600,R0
         CMPB    R0,#'z            ; русская буква "З"
         BEQ     VV4
         CMPB    R0,#'r            ; русская буква "Р"
         BEQ     VV5
         CMPB    R0,#'t            ; конец режима ?
         BNE     VV3
         RTS     PC
VV3:     CMPB    R0,#35            ; недопустимый код ?
         BGT     STOP              ; да--->STOP
         MOVB    XEH(R0),R0        ; R0-порядковый номер (1...14) клавиши
         BEQ     STOP              ; недопустимая клавиша ? да--->STOP
;
; Управление манипулятором
;
         ASL     R0                ; R0-индекс в таблицах XEH1,XEH2
         MOV     XEH1-2(R0),R2     ; R2-код управления манипулятором
         MOV     XEH2-2(R0),R1     ; R1-адрес платы И2 для вывода кода
         MOV     R2,@R1            ; вывод кода управления
         BR      VVOD              ; на опрос клавиатуры
;
; Управление патроном станка
;
VV4:     MOV     #167762,R1
         BIS     #2000,@R1
         BR      VVOD
VV5:     MOV     #167762,R1
         BIS     #1000,R1
         BR      VVOD
;.......................................................................
;  Таблица перекодировки кода клавиши в её порядковый номер          .
;  на функциональной клавиатуре                                      .
;.......................................................................
XEH:     .BYTE 0,14.,11.,0,0,0,0,0,5
         .BYTE 0, 0, 0, 0, 3, 0, 0, 0
         .BYTE 0,12.,9.,7.,1.,10., 0, 0
         .BYTE 6,4,13.,2,8.
;.......................................................................
;            Таблица кодов для вывода на платы И2                     .
;.......................................................................
XEH1:
;                B+    C+     B-
         .WORD     1, 120  ,  2
;                A-   заж.дет. A+
         .WORD     5, 10000,  6
;              заж.заг. C-   разж.заг.
         .WORD 40000,   140,100000
;                Д- разж.дет. Д+
         .WORD   240, 20000, 300
;                ротация схватов
         .WORD     4,    10
;.......................................................................
;                Таблица адресов плат И2                              .
;.......................................................................
XEH2:
;                B+     C+     -B
         .WORD 167762,167772,167762
;                A-   заж.дет.  A+
         .WORD 167772,167772,167772
;              заж.заг. C-  разж.заг.
```

```
        .WORD 167772,167772,167772
;                Д-  разж.дет.  Д+
        .WORD 167762,167772,167762
;               ротация схватов
        .WORD 167762,167762
;*************************************************************
;       Текст подсказки для вывода на экран 15ИЭ00-013        *
;*************************************************************
TEXT:   .BYTE   7,7,7,14,10,37,16
;  тексты условно показаны в русском регистре
.ASCII/РЕЖИМ РУЧНОГО УПРАВЛЕНИЯ/<12><15>
.ASCII/┌──────────┬──────────┬──────────┐/<12><15>
.ASCII/│ ПОВОРОТ  │ ОТ СЕБЯ  │ ПОВОРОТ  │/<12><15>
.ASCII/│ ──────   │    A     │    A     │/<12><15>
.ASCII/│    !     │    !     │    !──── │/<17><12><15>
.ASCII/│  V  B+   │    ! C+  │    B-    │/<16><12><15>
.ASCII/└──────────┴──────────┴──────────┘/<12><15>
.ASCII/           │  ЗАЖИМ   │          │/<12><15>
.ASCII/  ВЛЕВО    │  . ДЕТ.  │  ВПРАВО  │/<12><15>
.ASCII/ <──────   │     .    │  ──────> │/<12><15>
.ASCII/     A-    │     .    │    A+    │/<12><15>
.ASCII/           │          │          │/<12><15>
.ASCII/ ЗАЖИМ     │ К  СЕБЕ  │ РАЗЖИМ   │/<12><15>
.ASCII/ ЗАГ.      │     │    │      ЗАГ.│/<12><15>
.ASCII/ <──────   │     │ C- │  ──────> │/<17><12><15>
.ASCII/           │     V    │          │/<16><12><15>
.ASCII/           │          │          │/<12><15>
.ASCII/  ВНИЗ     │  РАЗЖИМ  │  ВВЕРХ   │/<12><15>
.ASCII/ ═══════   │  ! ДЕТ.  │    A     │/<17><12><15>
.ASCII/    ! Д-   │     V    │    ! Д+  │/<12><15>
.ASCII/     V     │     *    │ ═══════  │/<16><12><15>
.ASCII/           │          │          │/<12><15>
.ASCII/AP2,C1-ВРАЩЕНИЕ СХВАТОВ,  З-ЗАЖИМ ПАТРОНА/<12><15>
.ASCIZ/P-РАЗЖИМ ПАТРОНА,T-КОНЕЦ РЕЖИМА РУЧНОГО УПРАВЛЕНИЯ/
.EVEN
.END
```

ввод управляющей программы (УП) для станка с ЧПУ в ОЗУ микроЭВМ робота с фотосчитывателя пульта ЧПУ;

вывод УП в режиме прерываний в пульт ЧПУ станка;

оперативное редактирование УП для станка с ЧПУ в ОЗУ микроЭВМ с помощью дисплея 15ИЭ-00-013;

вывод откорректированной УП на перфоленту с помощью перфоратора ПЛ150;

тестирование механизмов робота и станка.

В качестве управляющей микроЭВМ РТК используется микроЭВМ МС1201.01 («Электроника 80»), для стыковки с устройством ЧПУ — параллельный байтовый интерфейс микроЭВМ МС1201. 01.

Максимальный объем УП для станка с ЧПУ, хранимой в ОЗУ МС1201.01—28 Кбайт.

Программное обеспечение имеет общий объем 6 Кслов (программа управления манипулятором — 4 Кслов, программы ввода-вывода перфоленты и редактора УП — 2 Кслов) и размещается в ПЗУ либо вводится в ОЗУ с перфоленты через фотосчитыватель станка с ЧПУ.

Перечень команд, выполняемых в режиме ручного управления РТК, приведен в табл. 1; в режиме редактирования УП с помощью экранного редактора текстов — в табл. 2.

В режиме редактирования УП блокируется ввод с клавиатуры символов, недопустимых в коде ISO-7.

Редактор содержит минимальный, но достаточный для выполнения оперативных работ, набор команд. При необходимости несложно включить дополнительные команды, расширяющие его возможности.

Передача управления из УП в редактор и обратно производится с помощью карты загрузки подпрограмм, хранящихся в ПЗУ. Это позволяет осуществить независимую замену версий отдельных функциональных частей ПО на микросхемах РПЗУ.

ПО состоит из мелких функциональных модулей на языке MACRO 11 и может быть легко адаптировано для РТК.

Тестовое программное обеспечение предназначено для проведения ремонтных и пусконаладочных работ. В его состав входят одиннадцать тестов:зажим и разжим патрона станка, проверка цикла обработки детали, проверка станка с ЧПУ, пошаговое выполнение цикла робота, проверка скорости движения механизмов робота, датчиков исходного положения, счетных датчиков, рабочей зоны механизмов, подсчет контрольных сумм микросхем РПЗУ.

На рисунке приведен листинг подпрограммы, обеспечивающей работу в режиме ручного управления манипулятором.

Таблица 2

Таблица соответствия символов клавиатуры командам редактора

Символ клавиатуры	Команда
Команды управления перемещением маркера на экране	
↓	Сдвиг вниз на одну строку
↑	Сдвиг вверх на одну строку
→	Сдвиг вправо на одну позицию
←	Сдвиг влево на одну позицию
ТАБ	Сдвиг вправо до следующего адреса в кадре УП
ГТ	Сдвиг влево до предыдущего адреса в кадре УП
↵ или ВК	Сдвиг в начало строки
ПС	Сдвиг в начало следующей строки
Редактирования	
↖	Поиск начала текста
ПРД	Поиск предыдущей экранной страницы (24 строки)
ПРС	Поиск следующей экранной страницы
ПРМ	Запрос модели для поиска
СТС	Поиск по модели
Перестановки и копирования участков текста, перенумерации кадров управляющей программы	
<	Указатель начала переставляемого или копируемого участка
>	Указатель конца переставляемого или копируемого участка
@	Указатель места, в которое нужно переставить или скопировать участок текста, ограниченный символами ⟨и⟩
СТР	Перестановка участка текста
АР1	Копирование участка текста
АР2	Перенумерация кадров (от 1 до 999 с шагом 1)

330064, Запорожье, ЗПОмоторостроитель, ОМАВ; тел. 61-43-17.

Статья поступила 13.05.88

УДК 621.865.8:681.3:06

В. С. Горбачев, В. Э. Липпинг

СИСТЕМА ТРЕХМЕРНОЙ МАШИННОЙ ГРАФИКИ ДЛЯ МОДЕЛИРОВАНИЯ ПРОМЫШЛЕННЫХ РОБОТОВ

В настоящее время за рубежом ведутся работы по созданию систем графической имитации гибких производственных модулей (ГПМ). Разработки в этой области выполняют такие известные фирмы, как IBM, General Electric, GMF (США), Mitsubishi, Hitachi (Япония), Nokia (Финляндия) и др.* Аналогичные работы ведутся и у нас в стране.

Рассмотрим систему трехмерной графики для моделирования и программирования робототехнических систем, реализованную в полном объеме на СМ4 и с ограниченными возможностями на ДВК2М с графическим дисплеем МС7401. Система позволяет: моделировать манипуляторы и другое технологическое оборудование, формируя модели на основе ограниченного набора стандартных фигур; изменять масштаб изображения и точки обзора; выводить на экран только интересующие исследователя элементы (например, схват, губки схвата с деталью и т. д.); удалять невидимые линии и т. д.

Укрупненная структура системы (рис. 1) состоит из баз данных объектов моделирования (ОМ) и ГПМ, модуля имитации, интерпретатора. Объектами моделирования являются манипуляторы, станки, другое технологическое оборудование, детали и оснастка.

Первоначально пользователь задает объекты моделирования, а затем компонует их в ГПМ. Далее эти модели автоматически заносятся в соответствующие базы данных, формируется управляющая программа и моделируется ее выполнение. Система позволяет обнаружить следующие конфликтные ситуации: столкновения, невозможность зажатия детали, несинхронность работы оборудования, нахождение программной траектории вне рабочей зоны манипулятора.

База данных ОМ обеспечивает хранение моделей ОМ и доступ к ним. Она состоит из набора библиотек (совокупность файлов, хранимых на магнитном диске).

При работе с моделями можно использовать функции базы данных: генерацию и коррекцию модели; уничтожение модели или библиотеки; изменение имени модели; копирование модели или библиотеки; распечатку списка имен библиотек, списка моделей в библиотеке, информации о модели, краткой инструкции по работе с базой данных; выход из модуля. При генерации ОМ пользователь указывает имя модели и имя библиотеки, в которую она будет включена. Далее он в диалоговом режиме задает требуемые параметры, создавая тем самым описание ОМ. В случае неверного ввода какого-либо параметра с помощью функции коррекции пользователь может изменить его.

База данных ГПМ. Возможности базы данных ГПМ во многом аналогичны возможностям базы данных ОМ.

При описании модели ГПМ пользователь последовательно определяет входящее в него оборудование, оснастку и детали, а затем указывает их начальное положение. Кроме того, он может изменить положение, уничтожить или добавить новый ОМ в модель ГПМ. Однако если при создании модели ОМ необходимо описывать все данные, то при описании ГПМ можно использовать уже существующую модель, модифицировать ее и получить тем самым новую. Модели ГПМ в файлах системы организованы по списочному принципу.

Интерпретатор. Входные данные для интерпретатора — команды управляющей программы. Он обеспечивает интерпретацию команд, осуществляет решение кинематических задач для манипуляторов, интерполирует траекторию, определяет время выполнения действий и т. д. Выходом интерпретатора является вектор обобщенных координат мани-

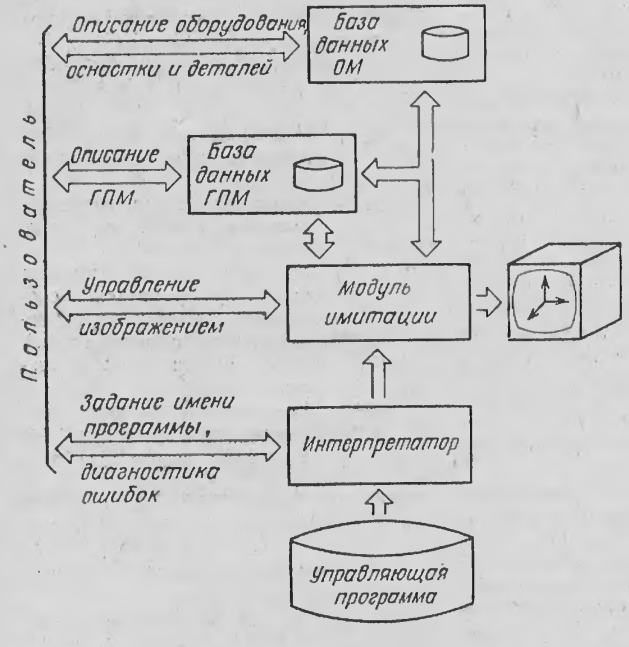


Рис. 1. Структура системы

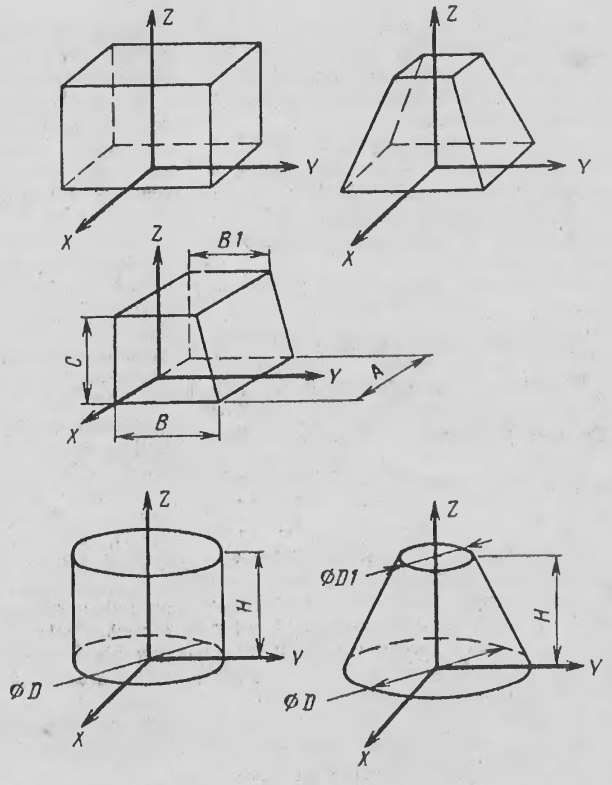


Рис. 2. Стандартные тела системы, размещение СКТ в них

* Горбачев В. С., Липпинг В. Э. Использование интерактивной машинной графики в промышленной робототехнике.— М.: ВНИИТЭМР, 1987.— 48 с.— (Технология, оборудование, организация и экономика машиностроительного производства. Сер. 9, Автоматизированные системы проектирования и управления: Обзорная информация; вып. 6).

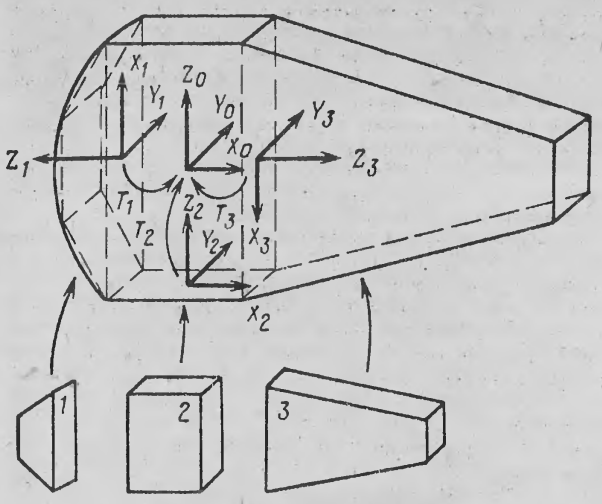


Рис. 3. Аппроксимация звена манипулятора РМ-01 стандартными телами

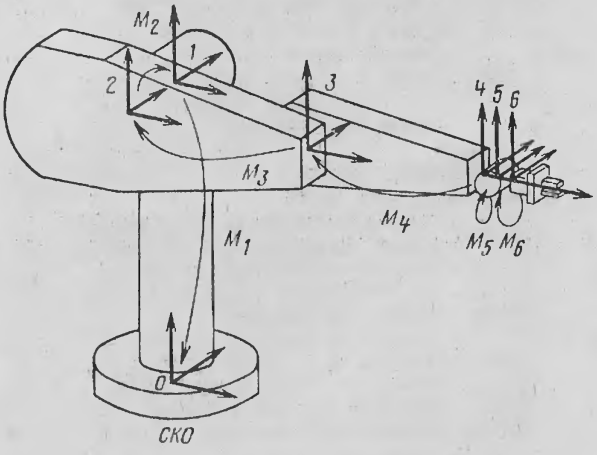


Рис. 4. Кинематическое описание манипулятора РМ-01

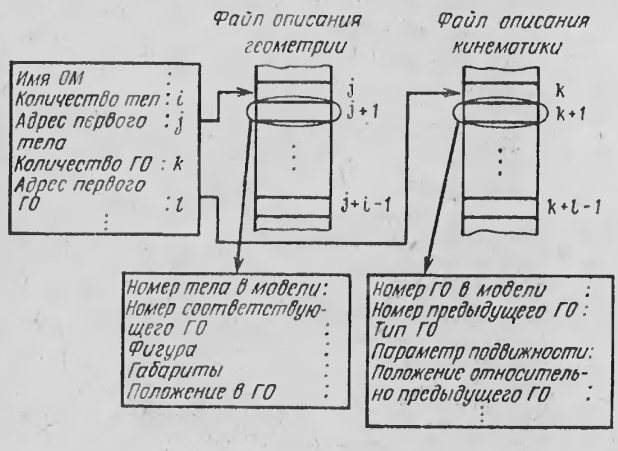


Рис. 5. Структура данных ОМ

пулятора и параметры, определяющие состояния схватов ПР для определенного момента времени.

Модуль имитации — основной модуль системы. Он включает в себя средства трехмерной машинной графики, которые определяют пространственные взаимодействия между оборудованием и деталями, а также создают изображения модели ГПМ в соответствии с данными, поступающими от интерпретатора.

Модуль функционирует в двух режимах: ожидания и непрерывной имитации.

В непрерывном режиме имитации на экране графического дисплея (ГД) создается «мультфильм» о работе ГПМ, а в режиме ожидания после создания очередного изображения система ожидает команды пользователя на продолжение имитации. Режим ожидания используется для изучения текущего состояния модели ГПМ, изменения точки обзора, масштаба, сдвига изображения или других графических параметров, а также при визуальном обнаружении столкновений. В данном режиме пользователь может заново прорисовать модель ГПМ с измененными графическими параметрами, что позволит более детально рассмотреть интересующую часть модели.

Изображение создается модулем в несколько этапов:

1) определение конфигураций манипуляторов и другого оборудования. Модуль рассчитывает положения в пространстве отдельных составных частей оборудования. Одновременно с этим автоматически проверяется принадлежность программной траектории рабочей зоне манипулятора;

2) определение положений деталей по состоянию схватов манипуляторов и информации, представленной в модели ГПМ. Решаются задачи взятия и освобождения деталей манипуляторами, контроля правильности захвата, помещения детали в заданную точку пространства;

3) отображение модели ГПМ на экране ГД. По известным положениям оборудования и деталей вычисляются их проекции на экране ГД в зависимости от заданной точки обзора, масштаба и других графических параметров.

Модель ОМ включает геометрическое и кинематическое описания. Геометрическое описание характеризует форму геометрических объектов (часть ОМ, которая остается неизменной). Например, объектом может быть звено манипулятора, состоящее из множества тел, которым аппроксимируется реальная форма. В качестве тел используются параллелепипед, прямая призма, пирамида, а также цилиндр и конус, поверхности второго порядка которых аппроксимированы плоскими гранями (рис. 2). С геометрическими объектами (ГО) связана система координат (СКГО), а с каждым телом ГО — система координат тела (СКТ). Например, положения тел в ГО описываются посредством матриц перехода Т из СКТ ((X_1, Y_1, Z_1), (X_2, Y_2, Z_2), (X_3, Y_3, Z_3)) в СКГО (X_0, Y_0, Z_0) (рис. 3).

Кинематическое описание содержит информацию о пространственном расположении ГО в виде матрицы перехода M_i из системы координат (СК) i-го ГО в СК другого ГО либо в базовую систему координат объекта (СКО) моделирования. Матрица M_i формируется из двух составляющих (рис. 4):

$$M_i = M_0 \times M_{vi},$$

где M_0 — матрица начального положения ГО; M_{vi} — матрица, определяющая изменение положения ГО по отношению к начальному.

В общем случае матрица M_{vi} является функцией от шести параметров: трех вращений и трех смещений. Однако принято, что пять из них постоянны или равны нулю, а значение шестого совпадает со значением обобщенной координаты.

Геометрические объекты могут быть следующих типов: неподвижные объекты; звенья манипуляторов; объекты, принимающие несколько фиксированных положений в пространстве; объекты, аппроксимирующие конвейерную ленту; губки захватных механизмов; связанные объекты, позволяющие моделировать механизм пантографа.

Для организации доступа к кинематическому и геометрическому описаниям манипулятора (рис. 5) используется набор данных, называемый каталогом, где содержатся имя ОМ, число СКГО и тел в нем, а также соответствующие ссылки. СКО в кинематическом описании всегда имеет порядковый номер.

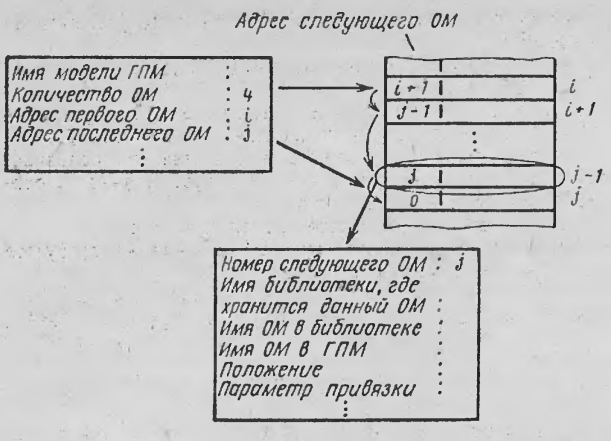


Рис. 6. Структура данных ГПМ

Модель ГПМ содержит информацию о количестве, компоновке и взаимодействии оборудования, оснастки и деталей на рабочем месте. Описания соответствующих ОМ к моменту генерации модели ГПМ должны присутствовать в системе. ОМ внутри ГПМ рассматривается как единое понятие.

Компоновка ГПМ начинается с выбора расположения некоторой базовой системы координат (БСК).

Затем пользователь определяет объекты моделирования, присваивает имена, по которым они будут идентифицироваться в модели рабочего места. Положение ОМ задается первоначально относительно БСК и может быть изменено с помощью соответствующей функции базы данных ГПМ.

На рис. 6 приведена структура модели ГПМ. Информация о каждом ОМ хранится в отдельных записях, организованных по списочному принципу.

Система графического моделирования реализована на языке Фортран 4 под управлением ОС RSX-11/M или в усеченной версии под управлением RT-11. Общий объем загрузочных файлов полной версии системы составляет около 130 Кбайт. Быстродействие системы — 1...10 с в зависимости от насыщенности отображаемой сцены.

Телефон 259-51-75, Москва

Статья поступила 26.02.88

УДК 681.3

А. Ф. Гадалов, Г. В. Борзов, Е. П. Ильин

СИСТЕМА РЕДАКТИРОВАНИЯ, ПЕРЕДАЧИ УПРАВЛЯЮЩИХ ПРОГРАММ И КОНТРОЛЯ ДЛЯ СТАНКОВ С УЧПУ «ЭЛЕКТРОНИКА НЦ-31»

В рамках системы КАПРИ разработана подсистема редактирования, передачи УП и контроля для УЧПУ «Электроника НЦ-31» на микроЭВМ семейства ДВК, которая включает в себя: ДВК, 4-канальный блок последовательного интерфейса «Телекварт» [1], блок дистанционного управления (БДУ) «Телепульт» [2].

Плата «Телекварт» с габаритными размерами 140×200 мм устанавливается в конструктив ДВК и обеспечивает связь с четырьмя устройствами «Телепульт» (возможно подключение до шести блоков). Если в конструктиве нет места для включения блоков «Телекварт», возможно расширение системной магистрали микроЭВМ за счет подключения дополнительного конструктива микроЭВМ «Электроника 60» при помощи кабеля с терминатором [3].

Блок «Телепульт» устанавливается в комплектном корпусе УЧПУ «Электроника НЦ-31» на место, зарезервированное под второй процессор УЧПУ, и не влияет на непосредственное выполнение функций УЧПУ.

Обмен между этими блоками осуществляется по последовательному асинхронному каналу ввода-вывода типа «токовая петля» со скоростью 9600 Бод.

По этому же каналу можно подключить ДВК к локальной сети КАПРИ [1].

Система редактирования, передачи УП и контроля написана на языке MACRO 11 и БИМС [4]. В зависимости от состава переферии ДВК система загружается через локальную сеть с HOST-компьютера (типа СМ4) с последующей работой в составе локальной сети (в случае отсутствия НГМД) или работает под управлением ОС ДВК автономно.

```
E31>H
СПИСОК КОМАНД:
                КОМАНДНЫЙ РЕЖИМ:
        R <FILE> - чтение УП с диска
        W <FILE> - запись УП на диск
        G        - чтение из УЧПУ (0,1,2,3,4 зоны)
        P        - запись в УЧПУ (1,2,3,4 зоны)
        CL       - очистка буфера редактора
        S <NUM>  - выход в экранный режим
                        со строки "NUM"
        Q        - конец редактирования

                ЭКРАННЫЙ РЕЖИМ:
        <DL>     - удаление строки
        <IL>     - вставка пустой строки перед маркером
        <ПС>     - вставка пустой строки после маркера
        <ЗБ>     - удаление символа слева от маркера
        <DC>     - удаление символа справа от маркера
        СТРЕЛКИ  - движение по стрелкам
        <ГТ>     - движение по табуляции вперед
        <ТАБ>    - движение по табуляции назад
        <ВК>     - движение в начало (конец) строки
        <СБР>    - регенерация текста на экране
        <HOME>   - выход в командный режим
```

Рис. 1. Команды экранного редактора

```
;===================
; СОСТ :СЕМЕНОВ
;-------------------
;ЗАКАЗ :
;  ПОЗ :
; ЧЕРТ :
; НАИМ :ШТУЦЕР
;-------------------
;ОПЕР.1:ПОДРЕЗАТЬ ТОРЕЦ
;ОПЕР.2:ЦЕНТРОВАТЬ
;ОПЕР.3:СВЕРЛИТЬ
;ОПЕР.4:НАРЕЗАТЬ РЕЗЬБУ
;-------------------
Z30000W
M00
M3
S5
F15
T1
M8
X2700*W
Z100W
Z0
X0
```

Рис. 2. Фрагменты редактирования УП, экранный режим

Возможности системы:

чтение УП из архивных зон УЧПУ и с дискеты в память микроЭВМ;

запись УП из памяти микроЭВМ в архивные зоны УЧПУ и на дискету;

редактирование УП в памяти микроЭВМ (рис. 1 и 2);

отображение состояния пульта УЧПУ на экране дисплея микроЭВМ (рис. 3);

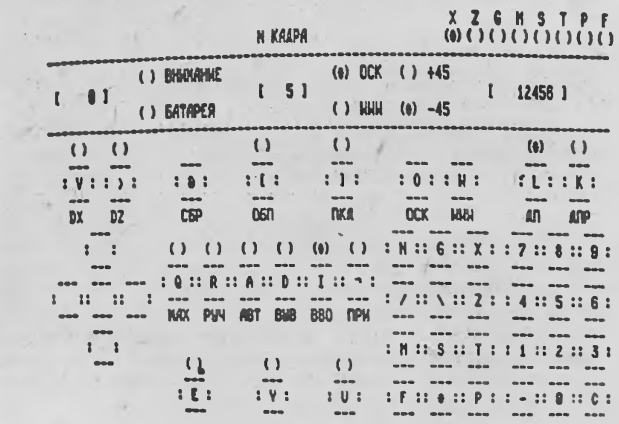


Рис. 3. Отображение пульта УЧПУ на экране дисплея микроЭВМ

Суммарное время работы станков за период с 07:00 18.05.88 по 07:00 19.05.88

Номер станка	Включен	Наладка	Работа по УП
Станок 1	10:15:24	03:45:16	06:21:50
Станок 2	00:00:00	00:00:00	00:00:00
Станок 3	14:10:05	00:00:00	13:11:02
Станок 4	10:12:34	01:02:11	08:00:02
Станок 5	07:20:25	01:02:15	04:21:22
Станок 6	12:22:35	06:01:00	05:14:24

Рис. 4. Таблица загрузки станков за отчетный период

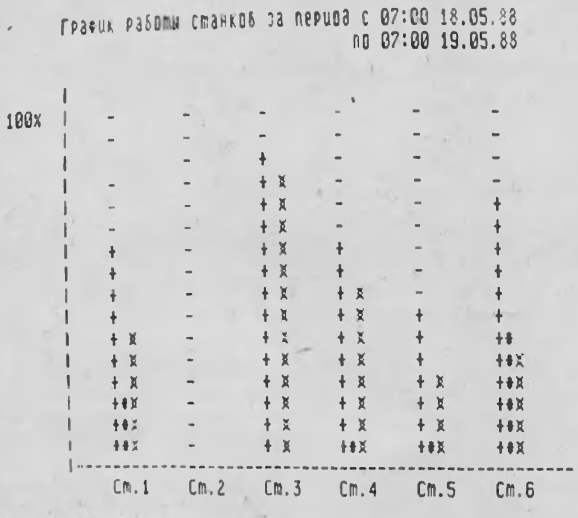


Рис. 5. График загрузки станков за отчетный период

дистанционное управление УЧПУ с клавиатуры дисплея микроЭВМ;

сбор статистики о станках по текущему состоянию пульта УЧПУ (рис. 4 и 5);

анализ оптимизированных оператором УП (сравнение режимов резания, подачи, последовательности обработки);

использование пульта УЧПУ в качестве специализированного терминала для связи оператора станка с подсистемами (вызов тестов, архивов);

связь с другими ЭВМ через локальную сеть.

Для выполнения своих функций системе не требуется специальных программ в УЧПУ.

123182, Москва, Д-182, пл. им. И. В. Курчатова, ИАЭ; тел. 196-97-53

ЛИТЕРАТУРА

1. Адамов Е. О., Иванов В. Г., Ляпунов М. М. Аппаратные средства КАПРИ // Микропроцессорные средства и системы.— 1987.— № 2.— С. 22—23.
2. Ильин Е. П., Ляпунов М. М. Аппаратура дистанционного контроля и управления с оперативными УЧПУ // Микропроцессорные средства и системы.— 1987.— № 2.— С. 41—42.
3. Электронная вычислительная машина «Электроника 60» 15ВМ-16: Техническое описание 2.791.004 ТО. — С. 35—37.
4. Ляпунов М. М. Интерпретирующая система БИМС.— Препринт ИАЭ-4291/15.— М.: ЦНИИатоминформ. 1986.— С. 33.

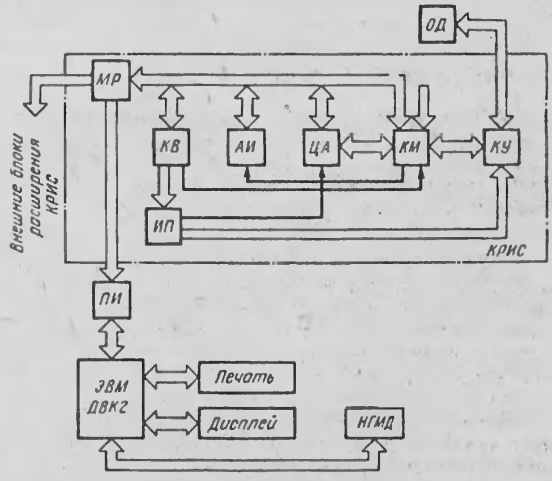
Статья поступила 22.08.88

УДК 621.3.049.75

С. М. Тоскин, В. В. Цатуров, В. С. Великанов, А. М. Афанасьев, И. А. Степанов, И. В. Бурочкин

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДИАГНОСТИРОВАНИЯ РАДИОЭЛЕКТРОННЫХ ИЗДЕЛИЙ КРИС

На кафедре «Радиотехнические устройства» Куйбышевского политехнического института им. В. В. Куйбышева на основе многомерных устройств сравнения разработана автоматизированная универсальная система диагностирования КРИС (см. рисунок).



Структурная схема изделия КРИС

КРИС состоит из функционально законченных модулей: комбинированных воздействий (КВ); аналоговых измерений (АИ); цифрового анализатора (ЦА); коммутатора (КМ), совмещенного с блоком установки режимов; контактного устройства (КУ) съема информации и источника питания (ИП).

Управляющие входы модулей через магистральный расширитель (МР) подключены к шине промежуточного интерфейса (ПИ). Протокол обмена МПИ преобразуется в протокол обмена периферийных БИС серии К580, на которых реализованы все модули системы. Собственных контроллеров модули не имеют.

Конструктивно специальный блок системы КРИС выполнен в стандартном корпусе АЦСКС-1024, выпускаемом промышленностью в качестве УСО для ЭВМ «Электроника Д3-28». В него устанавливается девять интерфейсных плат размерами 280×220 мм. Первую позицию в крейте занимает магистральный расширитель. Внутренняя магистраль системы выполнена на разъемах ГРПМШ-1-61.

По мнению разработчиков, наиболее эффективен способ многомерного сравнения с цифровой моделью при одновременной декомпозиции структуры объекта диагностирования (ОД) — это набор матриц связностей контактных точек, соответствующих характерным состояниям объекта.

Матрицы связностей ОД формируются в памяти ЭВМ либо путем считывания параметров связностей с образцового объекта, установленного на контактные штыри модуля КУ, либо аналитическим путем по принципиальной схеме объекта с учетом допустимых отклонений всех комплектующих элементов. При диагностировании цифровых схем ОД КРИС допускает использование тестовых способов, таблиц истинности и функциональный контроль электронных изделий, однако возможности системы ограничиваются техническими характеристиками составляющих модулей.

Технические характеристики

Число контактных точек при диагностировании	48
Вид диагностируемых изделий	Аналоговые, цифровые, комбинированные
Время оценки состояния годного изделия (100 ед. комплектующих), с, не более	60
Время поиска, с, не более	
одиночной неисправности	120
групповых неисправностей	300
Время перестройки системы на другое изделие без учета смены контактной группы и при наличии программы диагностирования, ч, не более	1
Габаритные размеры, мм	355×495×220

Ориентировочная стоимость изготовления специального блока КРИС 4—5 тыс. руб.

Разработаны комплект конструкторской документации системы; программное обеспечение для модулей; пакет тестовых программ настройки, регулировки и ремонта; пакет унифицированных программ диагностирования. Продолжается разработка ПО КРИС в среде операционной системы РАФОС на языке Паскаль.

443010, Куйбышев, ул. Галактионовская, 141, КПИ, кафедра РТУ, тел. 37-08-80, 37-04-37

Статья поступила 20.04.88

УДК 621.9.06-529

В. М. Водовозов, Ю. А. Горячев, А. К. Пожидаев

КОМПЛЕКТНАЯ СИСТЕМА ЧПУ НА БАЗЕ ДВК

Предлагаемая система ЧПУ СКАН-5 отличается от серийных систем компактностью, повышенной надежностью и простотой в обслуживании и программировании. Она предназначена для управления 3—4-координатным технологическим оборудованием — станками, роботами, а также движением робота в функции текущего положения одной из координат обслуживаемого станка.

Система выполняет функции автоматической отработки управляющей программы, подготовки программ непосредственно у станка, совмещения режимов подготовки и обработки программ, обучения роботов контурному формообразованию с автоматическим формированием программ в процессе обучения, прямого цифрового управления всеми приводами оборудования и ручного управления.

Программное обеспечение (64 Кбайт) состоит из базового и 50 управляющих программ общей длиной до 500 кадров, систематизированных в каталоге. Силовые вентили тиристорных преобразователей, собранные в четыре трехфазные мостовые реверсивные схемы, также входят в состав системы ЧПУ, придавая ей комплектность, завершенность, готовность к непосредственному монтажу на станке. При этом объем системы ЧПУ не превышают 1 м^3 вместе с дисплейным модулем, в качестве которого может использоваться любой из серийных комплексов ДВК1 — ДВК4, и выносным пультом обучения и управления.

Ввод ПО в ОЗУ при использовании ДВК1 осуществляется с фотосчитывающего устройства (ФСУ), а для вывода управляющих программ служит ленточный перфоратор (рис. 1). При использовании ДВК2 — ДВК4 для этих целей служат накопители на гибких магнитных дисках. Возможно хранение ПО в ПЗУ без использования внешних программоносителей.

Основа блока управления приводами (БУП) — четыре цифровые системы импульсно-фазового управления (ЦСИФУ), вырабатывающие импульсы управления тиристорами управляемых преобразователей (рис. 2). Ин-

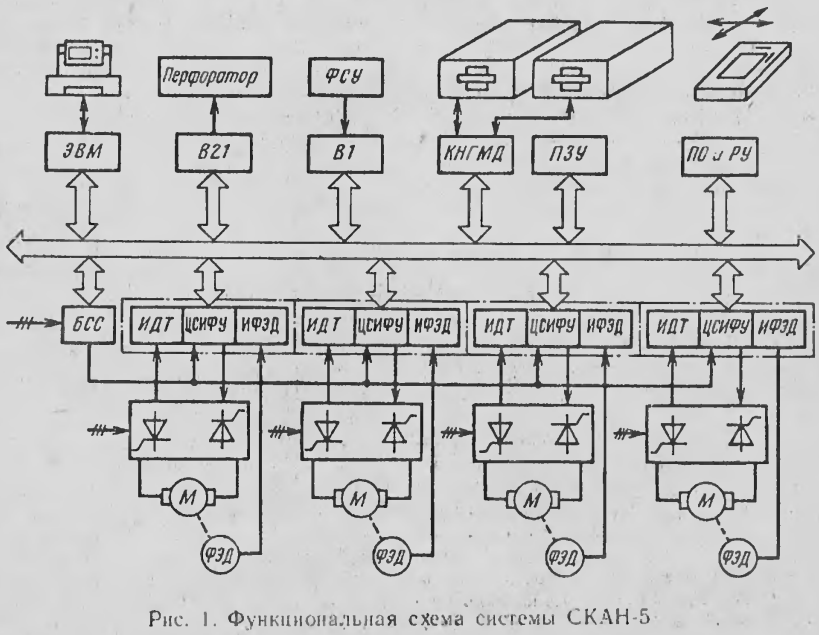


Рис. 1. Функциональная схема системы СКАН-5

тервальный таймер (КР580ВИ53) работает в режиме 4. Режим работы его каналов выбирается командами ЭВМ через шинный формирователь ШФ под управлением дешифратора адреса (ДША) и устройства управления (УУ). Угол управления тиристорами может меняться по каждому прерыванию в диапазоне 5...150°. Для согласования однополярной регулировочной характеристики ЦСИФУ с реверсивным преобразователем используются триггер В/Н и устройство переключения (УП). При изменении полярности выходного напряжения формирователь паузы (ФП) создает бестоковую паузу длительностью 6 мс. Если после отсчета выдержки времени тиристоры продолжают находиться в проводящем состоянии, ФП перезапускается по сигналу датчика проводимости (ДП). Во время бестоковой паузы блокируются управляющие импульсы. Если же во время паузы триггер В/Н возвращается в исходное состояние, ФП блокируется — обеспечивается высокое быстродействие и устойчивость работы преобразователя. Управляющие импульсы с формирователя и сигналы от блока синхронизации с сетью (БСС) поступают на распределители импульсов (РИ). Здесь осуществляется синхронизация и удваивание импульсов, которые через оптронные развязки подаются через усилители К, собранные по схеме составного транзистора, на управляемые вентили (УВ).

Интерфейс датчика тока (ИДТ) каждой координаты включает в себя интегратор (И), аналого-цифровой преобразователь (АЦП), подключенный через делитель (Д) к генератору (Г), и буферный регистр датчика тока (БРДТ). Датчик тока построен на базе трансформатора тока. Период перезапуска АЦП — 100 мкс; диапазон измерения тока — 1...256 А.

Интерфейс фотоэлектрических датчиков (ИФЭД) преобразует сигналы датчика в параллельный код и передает его в ЭВМ. Он состоит из формирователя импульсов (ФИДП), накопителя С4 и буферного регистра (БРДП).

Модуль БСС вырабатывает сигналы

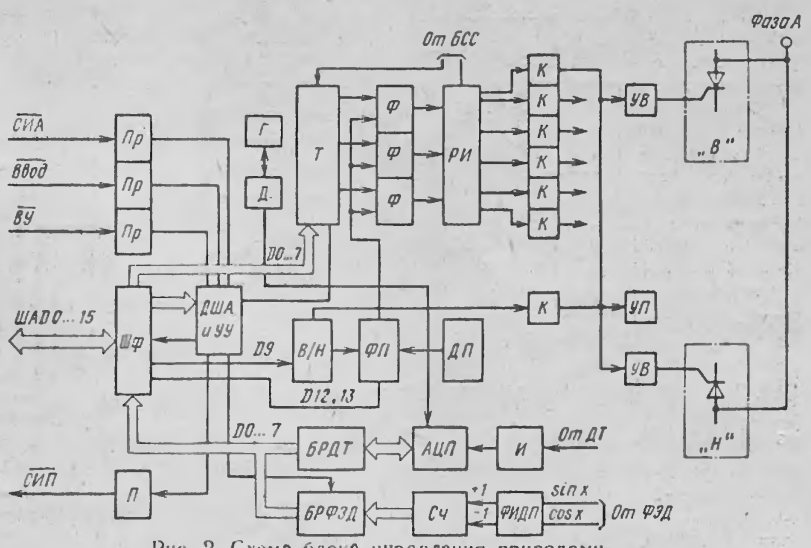


Рис. 2. Схема блока управления приводами

прерывания с периодом 3,3 мс, управляющие сигналы для работы ЦСИФУ, а также передает в ЭВМ сигналы от электроавтоматики станка. Так же, как и БУП, БСС связан с магистралью ЭВМ канальными передатчиками (П) и приемниками (Пр), шинными формирователями (ШФ) и ДША с УУ. Компараторы фиксируют переход фазного напряжения через нуль и определяют точку естественного открывания вентилей.

Выносной пульт обучения (ПО) и ручного управления (РУ) содержит четыре потенциометра, преобразующих угол поворота рукоятки в напряжение ±5 В. Сигналы передаются через четыре аналого-цифровых преобразователя К1113ПВ1 в буферные регистры под управлением генератора с периодом 60 мкс.

В системе предусмотрены два режима подготовки программ методом обучения. В соответствии с основным режимом оператор командами с ПО и РУ перемещает схват робота вдоль заданной траектории формообразования. Информация о положении рабочих органов заносится в память системы, где автоматически разбивается на кадры управляющей программы. Другой режим обучения предназначен для дополнения ранее подготовленной управляющей программы недостающей информацией.

Для реализации совместного движения робота и обслуживаемого им станка базовое ПО системы оснащено специальной программой функциональной интерполяции. В соответствии с этой программой задание на движение робота передается только после получения информации о движении ведущей координаты станка. Тем самым осуществляется геометрическая взаимосвязь координат станка и робота. При неподвижном станке эта связь прерывается, и робот получает возможность автономных движений.

Система ЧПУ СКАН-5 проходит опытно-промышленную эксплуатацию, результаты которой определят перспективу развития комплексных систем подобного класса.

Телефон 234-89-64, Ленинград

Статья поступила 20.06.88

СПРАВОЧНАЯ ИНФОРМАЦИЯ

УДК 681.3.01:51

РЕКОМЕНДАЦИИ ПО ПРИМЕНЕНИЮ МИКРОСХЕМ СЕРИИ К1801

(Окончание цикла. Начало в № 4—6, 1988, № 1, 1989)

1. Аппаратура, сконструированная на основе микропроцессора К1801ВМ1, будет гарантирована от сбоев в работе, если использовать «схемы поддержки», выполненные с применением микросхем серий К155, К555, К530 (рис. 1—5). Резервные выводы микропроцессора через резисторы 2...3кОм подключаются к источнику напряжения питания 5 В.

2. Примеры применения микросхемы К1801ВП1-030 для управления блоками памяти 16К × 16 и 32К × 16 иллюстрируются на рис. 6 и 7 соответственно.

3. Совместное применение микросхем К1801ВП1-033 и К1801ВП1-034 для построения интерфейсов параллельного ввода-вывода и байтового параллельного ввода-вывода, устройства выдачи вектора прерывания и компаратора адреса представлены на рис. 8—10 соответственно.

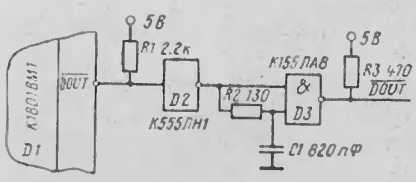


Рис. 1. Схема поддержки сигнала DOUT микропроцессора К1801ВМ1

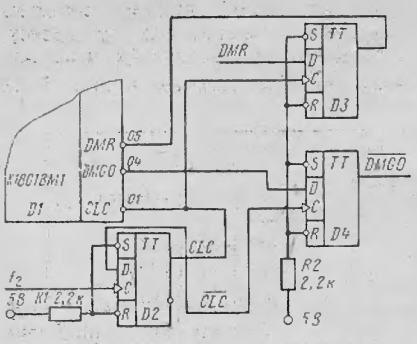


Рис. 2. Схема поддержки режима прямого доступа в память микропроцессора К1801ВМ1: D2...D4 — К155ТМ2,

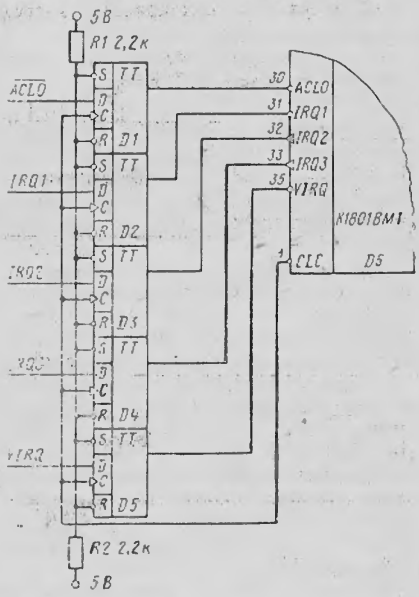


Рис. 3. Схема поддержки прерываний микропроцессора К1801ВМ1: D1...D5 — К155ТМ2

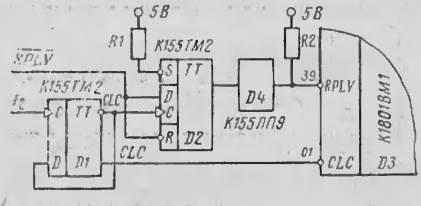


Рис. 4. Схема поддержки сигнала RPLY микропроцессора К1801ВМ1

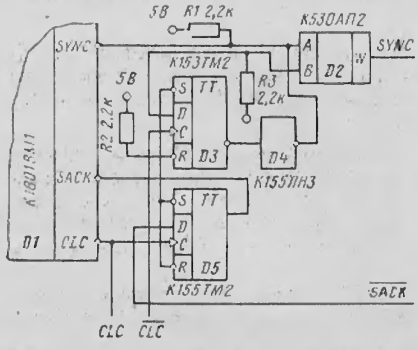


Рис. 5. Схема поддержки сигналов SYNC и SACK микропроцессора К1801ВМ1

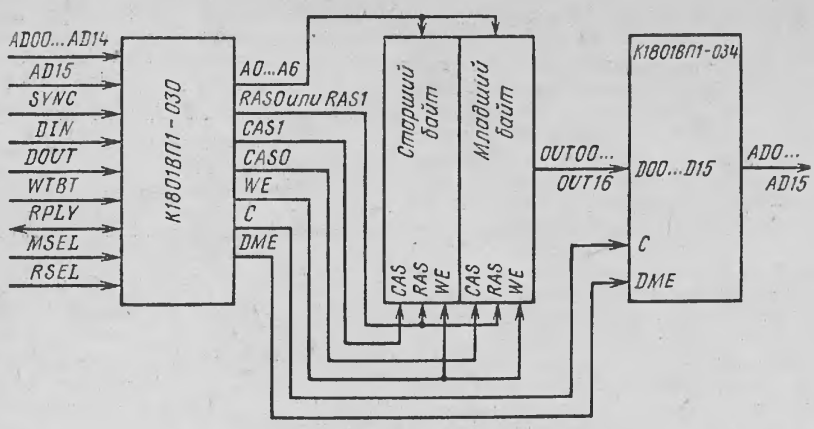


Рис. 6. Схема управления блоком ОЗУ 16К 16-разрядных слов: если AD15=1, то RAS0; если AD15=0, то RAS1

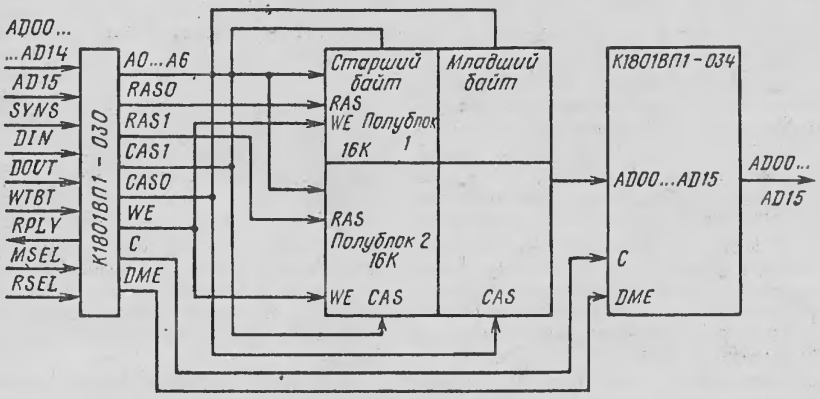


Рис. 7. Схема управления блоком ОЗУ

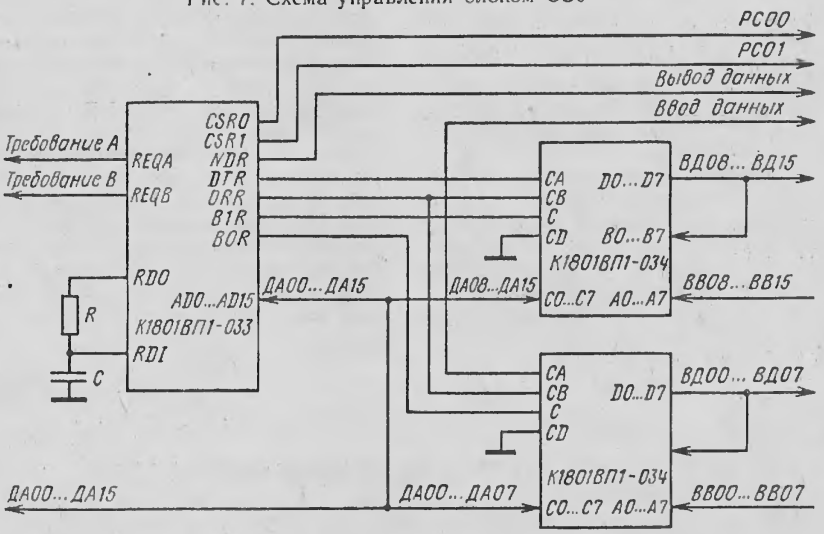


Рис. 8. Интерфейс параллельного ввода-вывода

4. Особенности применения микросхемы К1801ВП1-035

Группы адресов регистров и источников адресов прерываний задаются комбинаций уровней на выводах ACL0 и ACL1 (табл. 1). При заданной ком-

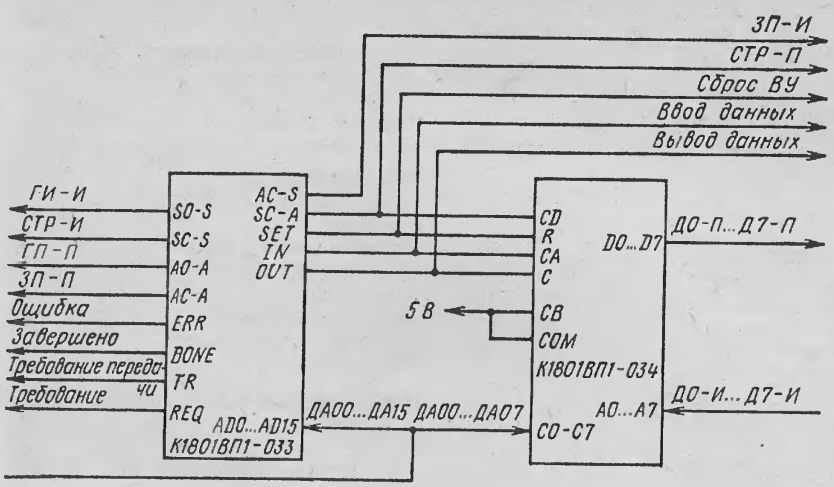


Рис. 9. Интерфейс байтового параллельного ввода-вывода

бинации ACL0=1, ACL1=1 регистры микросхемы выбираются по адресу, состоящему из разрядов 1,2, и по сигналу $\overline{BS}$. На разряды адреса 3...12

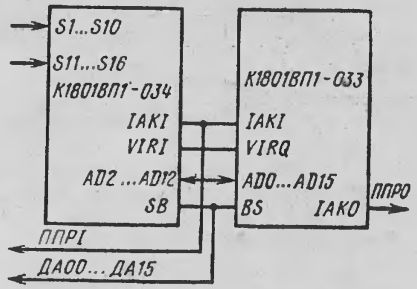


Рис. 10. Устройство выдачи вектора прерывания и компарации адреса

Таблица 1

Уровни сигналов на входах ACL0, ACL1

Обозначение	Вывод	Группа			
		1	2	3	4
ACL0	23	0	1	0	1
ACL1	24	0	0	1	1

Примечание. 0,1 — низкий и высокий уровни на входе.

Таблица 2

Уровни сигналов на входах NB0, NB1

Обозначение	Вывод	5 бит	7 бит	8 бит
NB0	7	X	0	1
NB1	8	0	1	1

Примечание. X — безразличное состояние.

Таблица 3

Уровни сигналов на входах FR0...FR3

Скорость обмена, Бод	FR3 (вывод 6)	FR2 (вывод 5)	FR1 (вывод 4)	FR0 (вывод 3)
50	0	0	0	0
75	0	0	0	1
100	0	0	1	0
150	0	0	1	1
200	0	1	0	0
300	0	1	0	1
600	0	1	1	0
1200	0	1	1	1
2400	1	0	0	0
4800	1	0	0	1
9600	1	0	1	0
19 200	1	0	1	1
57 600	1	1	0	0

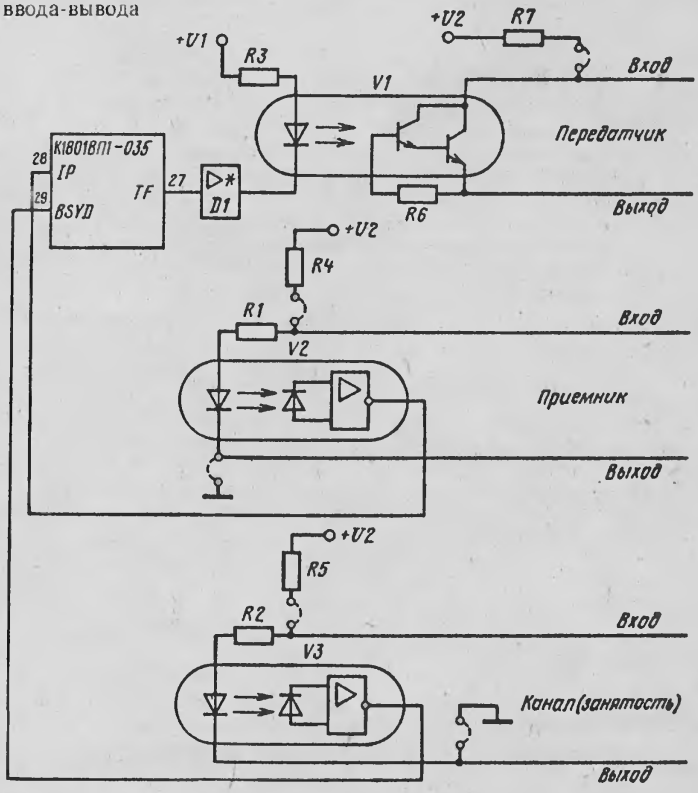


Рис. 11. Схема сопряжения микросхемы К1801ВП1-035 с токовыми петлями: V1 — оптрон АОТ1101; V2, V3 — оптроны К290ЛП1; D1 — микросхема с открытым коллектором; U2 — источник тока в петлях; R — токозадающие резисторы

микросхема не реагирует. Группа 4 адресов может использоваться для внешнего произвольного набора адресов регистров и внешнего источника адресов векторов прерываний.

Форматы посылок по последовательному каналу задаются комбинацией уровней на входах NB0 и NB1 (табл. 2), скорость обмена — на входах FR0...FR3 (выводы 3...6) (табл. 3).

Режим работы с паритетом или без него, а также с паритетом по четности или нечетности выбирается в зависимости от комбинации уровней на входах NP и PEV (выводы 30 и 32) (табл. 4).

Состояния выхода TF передатчика в зависимости от уровня на входе WSYD (вывод 29), наличия Лог. 1 в разряде 0 регистра состояний (PC) передатчика и подачи сигнала PCLO показаны в табл. 5.

Микросхема имеет выходы HALT (вывод 31), VIRQ (вывод 35) и RPLY

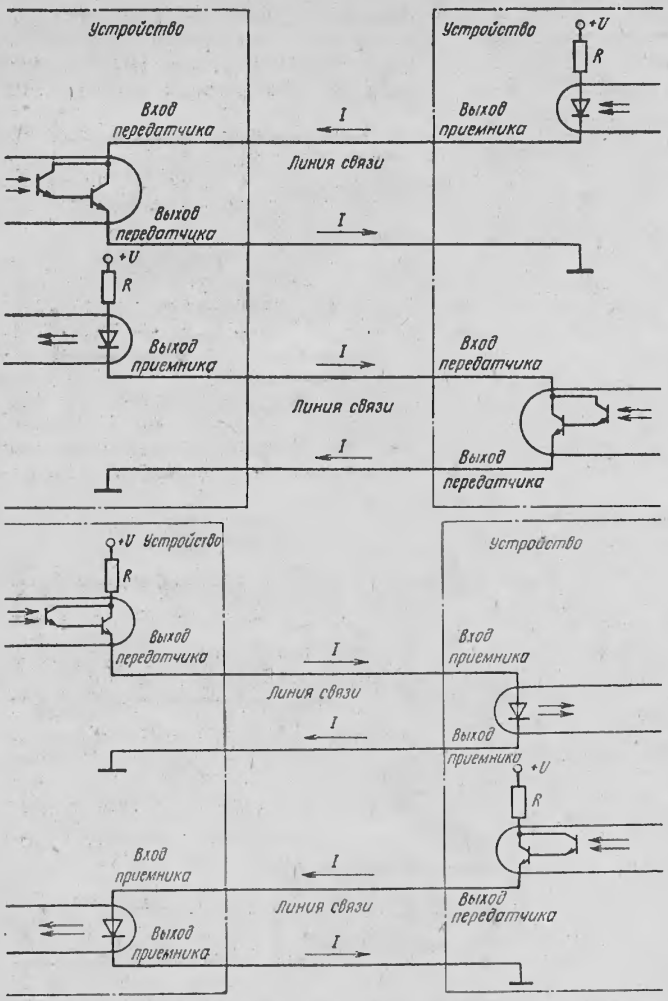


Рис. 12. Схема организации токовых петель при активных приемниках и пассивных передатчиках и наоборот

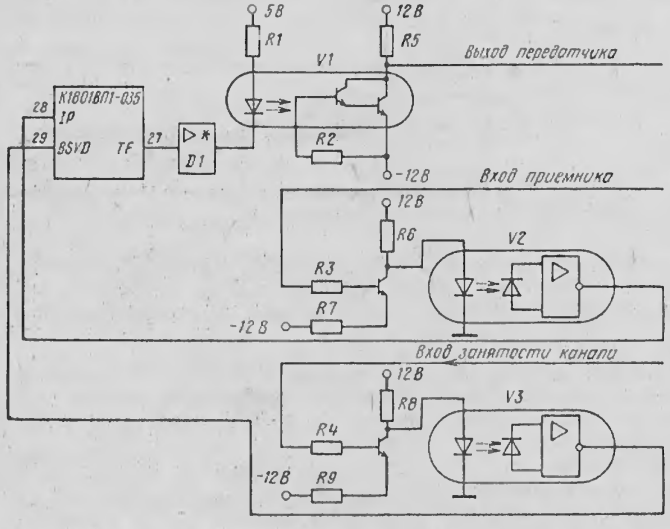


Рис. 13. Схема сопряжения микросхемы К1801ВП1-035 с телеграфным интерфейсом 12 В:
V1 — оптрон АОТ110А; V2, V3 — оптроны К293ЛП1; D1 — микросхема с открытым коллектором

Таблица 4

Уровни сигналов на входах NP и PEV

Режимы работы с паритетом	NP (вывод 30)	PEV (вывод 32)
Нет формирования бита паритета и контроля паритета	1	X
Формирование бита четности и контроль четности	0	1
Формирование бита нечетности и контроль нечетности	0	0

Таблица 5

Состояние выхода TF передатчика

Режим работы микросхемы и состояние линии	Вход BSYD	Наличие Лог 1 в разряде 0 РС передатчика	Состояние выхода TF передатчика
Выдача посылки, линия свободна	0	Нет	Выдача посылки
Линия занята	1	X	0(СТОП)
«Разрыв» линии	0	Есть	1 (СТАРТ)
Подача сигнала $\overline{DCL0}$	X	X	0 (СТОП)

(вывод 39) с открытым стоком, обеспечивающим только низкий уровень сигнала. Для формирования на этих выходах высокого уровня необходимо между ними и шиной источника питания 5 В включить резистор с номинальным значением, рассчитанным, исходя из величины выходного тока. Лог.0 I_{OL}, равного не более 3,2 мА.

Начальная установка микросхемы осуществляется подачей сигналов низкого уровня на входы INIT (вывод 34) и DCLO (вывод 40). Длительность сигналов на входе INIT — не менее 10 мкс, на входе DCLO — не менее 5 мс.

Для работы микросхемы со специальными последовательными линиями: токовыми петлями, телеграфным интерфейсом и другими, — необходимы специальные схемы поддержки. Сопряжение микросхемы с токовыми петлями (рис. 11) позволяет обеспечить ток 20 или 40 мА. При организации токовых петель схемы сопряжения, содержащие источники питания, являются активными, иначе,— пассивными. Организация токовых петель при активных приемниках и пассивных передатчиках, при активных передатчиках и пассивных приемниках представлена на рис. 12. Сопряжение микросхемы с телеграфным интерфейсом показано на рис. 13.

Г. Г. Глушкова

УДК 681.323

Шмат В. К., Усов Н. Н. **БИС форматора данных для управления СБИС ЗУ ЦМД** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 2.

Описан минимальный набор БИС управления, включающий контроллер, форматор данных и блок защиты информации, достаточный для построения систем памяти на СБИС ЗУ ЦМД.

УДК 621.327

Лангуев В. В., Болотов С. А., Ольшак А. И. **Комплект БИС серии КМ1823 для микропроцессорных систем управления двигателями внутреннего сгорания** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 12.

Рассмотренный комплект позволяет создавать электронные блоки (контроллеры) микропроцессорных систем, обеспечивающие управление углом зажигания в зависимости от частоты вращения коленчатого вала двигателя, нагрузки, температуры охлаждающей жидкости и управление топливоподачей в режиме принудительного холостого хода.

УДК 681.3

Крюков В. А. **Анализ принципов объектно-ориентированного программирования** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 14.

Рассматривается объектно-ориентированное программирование (ООП) на базе новой парадигмы программирования. Понятие объекта соответствует общедисциплинарному понятию система, что облегчает постановку и решение задач на ЭВМ в самых разных предметных областях. Дан анализ влияния новой парадигмы на язык, систему и технологию программирования. Для иллюстрации основных механизмов ООП используется язык-система Small Talk-80.

УДК 681.3

Абрамов С. М., Пименов С. П., Абакумов А. А., Хаткевич М. И. **Компьютерный вирус** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 22.

Рассматривается опыт борьбы с программой-вирусом DOS-62. Описываются разработанные в ИПС АН СССР программные средства диагностики и лечения инфицированных программ и защиты от программ-вирусов ПЭВМ класса IBM PC XT/AT.

УДК 681.3

Акулов В. К., Журавлев В. И., Новожилов Н. А., Шинкевич С. Л. **Система управления реляционными базами данных** // Микропроцессорные средства и системы.— 1989.— № 1.— С. 24.

Рассматривается система управления реляционными базами данных RDBMS для ОС RT-11 TSX$^+$. Система не критична к объему памяти ЭВМ, имеет простой и удобный язык формулирования запросов к БД. Обеспечивает программный интерфейс к БД для языков высокого уровня.

УДК 658.5.011.56

Вельтмандер П. В., Жуков Г. В., Килина Л. В., Малярчук А. М., Минин В. Ф. **Системы автоматизированного конструирования для персонального АРМ** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 46.

Рассматриваются функциональные возможности и организация диалога двух САПР. Приводится состав технических средств АРМ конструктора. На основе опыта эксплуатации этих САПР делаются выводы о дальнейшем их применении и развитии.

UDC 681.323

Shmat V. K., Usov N. N. **Data formatter LSI for bubble memory control.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 2.

The minimum subset of LSI for magnetic bubble memory support includes controller, data formatter and error corrector. The subset is sufficient for building controllers in mass storage units using bibble-memory VLSIs.

UDC 621.327

Languyev V. V., Bolotov S. A., Olshak A. I. **KM1823 LSI family for microprocessor control of combustion motors.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 12.

A microprocessor LSI family inteded to be used in microprocessor controllers of car motors is described. The ICs perform ignition angle control as a function of rotation rate, mechanic load, temperature of cooling liquid and regulate fuel flow in a loadless mode of the motor.

UDC 681.3

Kryukov V. A. **The analysis of object-oriented programming.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 14.

The object-oriented programming (OOP) method based on new programming paradigma is examined. The 'Object' means 'System' in conventional sense and helps to formulate and solve tasks in various discipline fields with the aid of computer. The influence of new paradigma on language, system and programming technique is analyzed. The illustration of main OOP mechanisms is given in SmallTalk-80 languade-system.

UDC 681.3

Abramov S. M., Pimenov S. P., Abakumov A. A., Khatkevich M. I. **Computer virus.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 22.

The practice of struggle agains virus program is discussed. The new software tools for virus detection and program treatment created in the IPS AN USSR for IBM PC XT/AT computers are described.

UDC 681.3

Akulov V. K. Zhuravlev V. I. Novozhilov N. A., Shinkevich S. L. **Relation Database management systems.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 24.

The manager of relation database contents RDBMS for RT-11 and TSX-Plus operating systems is described. The database management system is non-critical to available memory size and has simple and easy-to-use data reguest language. The software support for high-level language programming is also provided.

UDK 658.5.011.56

Weltmander P. V., Zhukov G. V., Kilina L. V., Malyarchuk A. M., Minin V. F. **CAD systems for personal workstation.** // Microprocessor devices and systems.— N. 2.— P. 46.

The functions and dialogue means of two CAD systems are discussed. The contents of hardware units of designer's workstation is listed. The practice of CAD systems usage as well as recomendations for their applications and future development are described.

УДК 681.325

Сайкин В. Т., Артюх И. Г., Карпов И. И., Юданов В. А. **Учебный робот** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 68.

Представлена новая модель учебного оборудования для организации начального обучения основам программирования, управления робототехническими устройствами и знакомства с элементами гибких производственных систем в общеобразовательной школе.

УДК 681.325

Макаров И..М. и др. **Учебная гибкая производственная система** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 71.

Рассмотрена структура аппаратных и программных средств учебной ГПС, состоящей из шести модулей: центра механообработки тел вращения, автоматизированного склада, транспортной тележки и трех роботизированных сборочных центров (специализированного, универсального и адаптивного). Общее управление режимами работы ГПС осуществляется от мини-ЭВМ Мера-60. Показана возможность широкого применения учебной ГПС в вузах для специальности «Робототехнические системы и комплексы».

УДК 681.3.01

Руцков М. В. **Мультипортовая память для построения видеокомплексов** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 77.

Изложены принципы комплексирования вычислительных средств с использованием мультипортовой памяти, как главного связующего узла системы. Приведены варианты построения интерфейсов вычислительных средств с применением механизмов расширения адресации, ориентированных на обработку изображений. Рассмотрена структура специализированного устройства сопряжения «Пиксел»

УДК 621.865.8:681.3.06

Горбачев В. С., Липпинг В. Э. **Система трехмерной машинной графики для моделирования промышленных роботов** // Микропроцессорные средства и системы.— 1989.— № 2.— С. 86.

Рассмотрена система графического моделирования промышленных роботов и робототехнических систем. Приведена структура программных средств и дано описание ее основных модулей. Показан способ представления моделей оборудования, прежде всего манипуляторов, на основе ограниченного набора простейших тел. Содержится информация о аппаратной реализации системы и ее быстродействия.

UDK 681.325

Saikin V. G., Artyuh I. G., Karpov I. I., Yudanov V. A. **Educational robot.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 68.

The new model of educational set of robotic units which may be used for primary study of programming, robotic control and flexible manufacturing system in public schools is described.

UDK 681.325

Makarov I. M. et al. **Educational flexible manufacturing system.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 71.

The hardware and software structure of modular edicational FMS is explained. The system include 6 modules: the machine-tool for mechanic processing of cgllindric bodies, automated store, transport unit and three robotic assembly centers (specialized, universal and adaptive ones). MERA-60 microcomputer controls general operation of FMS. The possibility of wide-spread usage of the FMS in high school for teaching robotics is displayed.

UDK 681.3.01

Rutskov M. V. **Multi-port memory for image processing** // Microprocessor devices and systems.— 1989.— N. 2.— P. 77.

The principles of computer system architecture using multi-port memory as main data interface are explained. The variants of system interfaces with memory mapping suitable for video image processing are described. The structure of "Pixel" — specialized videointerface is shown.

UDC 621.865.8:681.3.06

Gorbachev V. S., Lipping V E. **3-D computer graphics system for industrial robot simulation.** // Microprocessor devices and systems.— 1989.— N. 2.— P. 86.

The system for graphic simulation of industrial robots and robotic systems is discussed. The structure of software tools is shown and operation of main program modules is explained. The method for description of devices (manipulators as typical applications) using limited set of primitives is described. The system hardware configuration and its speed benchmarks are also illustrated.

Номер подготовили:
Е. И. Бабич, Г. Г. Глушкова,
В. М. Ларионова, С. С. Матвеев
Корректор Е. М. Кучерявенко
Техн. редактор Г. И. Колосова
Адрес редакции журнала:
103051, Москва, Малый
Сухаревский пер., д. 9-а
Тел.: 208-73-23; 208-19-94

Сдано в набор 18.01.89. Т—05084
Подписано к печати 20.03.89
Формат 84×108¹/₁₆
Офсетная печать.
Усл. печ. л. 10,08. Уч.-изд. л. 14,8
Тираж 108 120
Заказ № 5125
Цена 1 р. 10 к.

Орган Государственного
комитета СССР
по вычислительной технике и
информатике

Набрано в ордена Трудового Красного Знамени Чеховском полиграфическом комбинате Государственного комитета СССР по делам издательств, полиграфии и книжной торговли. 142300, г. Чехов, Московской обл.

Отпечатано в Московской типографии № 13 ПО «Периодика» Государственного комитета СССР по делам издательств, полиграфии и книжной торговли. 107005, г. Москва, Денисовский пер., 30. Заказ 76.



Универсальный сборочный центр



Адаптивный сборочный центр

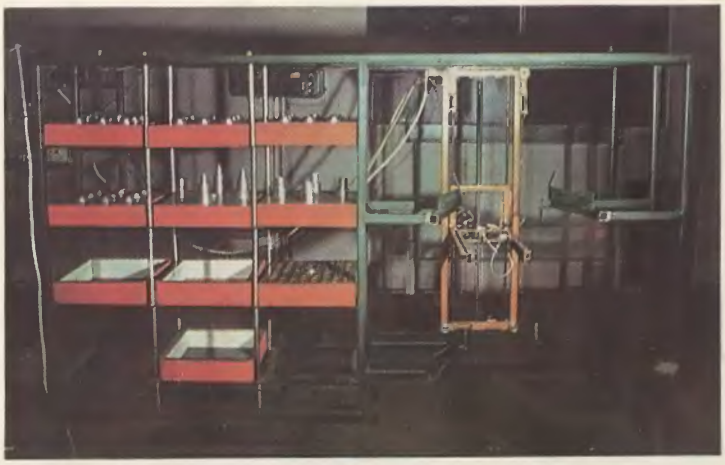
(Окончание.
Начало на 2 с. обл.)

УЧЕБНАЯ ГПС

Общее управление основными режимами работы учебной ГПС и ее отдельными модулями осуществляется мини-ЭВМ МЕРА-60 с пятью дисплеями.

Базовое методическое и программное обеспечение, разработанное также в МИРЭА, предусматривает решение на учебной ГПС около 30 различных задач в рамках лабораторного практикума курсового проектирования, которое может быть значительно расширено учащимися в процессе самостоятельной работы.

Учебная ГПС — удобный полигон для внедрения результатов студенческих работ; она обеспечивает возможность выполнения и реализации групповых проектов, что развивает способности студентов к коллективной работе над единой темой.



Автоматизированный склад



Центр механообработки тел вращения

(к ст. Макарова И. М. и др.)

1р. 10к. Индекс 70588

НАУЧНО-ПРОИЗВОДСТВЕННЫЙ ЦЕНТР «ИНФОРМАТИКА»

совместное советско-болгарское предприятие в области компьютерных технологий и систем

НПЦ «ИНФОРМАТИКА» выполняет следующие виды услуг:

разработку аппаратно-программных средств распределенных проблемно-ориентированных вычислительных комплексов и систем «под ключ», дружественных к пользователю;

поставку систем на базе различных типов ЭВМ;

обучение пользователей;

сервисное обслуживание техники и сопровождение систем.

Помимо перечисленных услуг **НПЦ «ИНФОРМАТИКА»** выполняет проектные работы, создает и поставляет заказчикам комплексы АСНИ, АСУ ТП, АСК организационных АСУ и других систем на базе распределенных комплексов ЭВМ различных типов.

В активе **НПЦ «ИНФОРМАТИКА»** создание и поставка недорогих локальных сетей ЭВМ со средой передачи данных в виде витой пары проводов, создание систем на базе локальных сетей; разработка информационно-измерительных комплексов с большим числом (до 1000) каналов измерения и управления; разработка генераторов тестовых воздействий, кросс-систем, гибридных вычислительных систем и т. п.

В области программных продуктов. **НПЦ «ИНФОРМАТИКА»** предлагает широкий набор пакетов программ для решения задач управления на базе персональных ЭВМ, обучающие системы для персональных ЭВМ и ЕС ЭВМ, оригинальную автоматическую систему учета и оперативного планирования ресурсов ЕС ЭВМ в среде СВМ, а также ряд программных систем на основе новейших версий баз данных персональных ЭВМ и ЕС ЭВМ.

В области обучения. **НПЦ «ИНФОРМАТИКА»** обучает пользователей работе на персональных компьютерах супермини-ЭВМ и и ЭВМ типа ЕС. Обучение ведется недельными циклами на основе интенсивных информационных технологий и методик.

Полный курс обучения работе на персональных компьютерах — шесть циклов. Подготовленный пользователь может быть зачислен на любой из них.

Цикл I — Операционная система MS-DOS, редактор, алгоритмический язык Бейсик.

Цикл II — Алгоритмический язык Паскаль (алгоритмический язык С). Решение инженерных и экономических задач.

Цикл III — Пакеты прикладных программ для решения инженерных и экономических задач

Цикл IV — Базы данных. Прикладное применение баз данных.

Циклы V и VI формируются как специализированные.

Для пользователей — инженеров:

Цикл Va — Машинная графика.

Цикл VIa — Основы автоматизации проектирования (по отраслям техники).



Для пользователей — управленцев:

Цикл Vб — Информационные системы.

Цикл VIб — Элементы АСУ. Учетные системы.

Аналогичным образом структурированы курсы для обучения пользователей супермини-ЭВМ и ЭВМ типа ЕС.

После каждого шестидневного цикла слушателям рекомендуется сделать перерыв на 2...4 недели для самостоятельного закрепления полученных знаний на производстве.

Занятия проводятся в виде семинаров. Каждый слушатель обеспечивается рабочим местом на базе персональной ЭВМ «Правец-16». По завершении занятий слушателям предоставляются информационные материалы по изучаемому курсу. Для иногородних занятия организуются на базе Подмосковного дома отдыха.

В области сервисного обслуживания. **НПЦ «ИНФОРМАТИКА»** осуществляет установку, гарантийный и послегарантийный ремонт ЭВМ производства НРБ.

Адрес НПЦ «ИНФОРМАТИКА»:
123459, Москва, ул. Новопоселковая, дом 6
Телефон для справок: 199-69-33
Заявки на обучение направлять по адресу:
125565, Москва, Ленинградское шоссе, дом 84
А/я 313