



МИКРО- ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

4 | 1985

ISSN 0233-4844

КАПРИ — интегрированная производственная система автоматизации проектирования, разработки и изготовления изделий машиностроения

Микропроцессорная система управления производственным комплексом из станков с ЧПУ, промышленных роботов, транспортных складов

Евромеханика — перспективная конструкционная система микропроцессорных средств

Эксплуатационная надежность и уровень сервиса — важнейшие характеристики эффективности внедрения персональных ЭВМ

Автономная настраиваемая кросс-система ПРА на базе ППЭВМ

Программа вузовского курса по специальности «Информатика и системное программирование» содержит ядро — курсы, составляющие обязательный минимум по алгоритмике и программированию

В номере: материалы к обсуждению проекта Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года



Высшей целью экономической стратегии партии был и остается неуклонный подъем материального и культурного уровня жизни народа. Реализация этой цели в предстоящем периоде требует ускорения социально-экономического развития, всемерной интенсификации и повышения эффективности производства на базе научно-технического прогресса. Предстоит поднять на качественно новую ступень производительные силы и производственные отношения, кардинально ускорить научно-технический прогресс, обеспечить быстрое продвижение вперед на стратегических направлениях развития экономики, создать производственный потенциал, равный по своим масштабам накопленному за все предшествующие годы.

Исходя из этого, в период до 2000 года:

...Существенно улучшить условия труда, добиваться более быстрого сокращения ручного труда и снизить его долю в производственной сфере до 15—20 процентов. Расширять возможности для развития и применения творческих способностей всех граждан.

...Перевести производство на преимущественно интенсивный путь развития, добиться кардинального повышения производительности общественного труда и на этой основе ускорить темпы экономического роста. Увеличить национальный доход страны почти в 2 раза.

Обеспечивать прирост национального дохода и продукции всех отраслей материального производства полностью за счет повышения производительности труда. Поднять производительность общественного труда в 2,3—2,5 раза, сделать решающий шаг в осуществлении программной задачи достижения по этому показателю высшего мирового уровня.

...Снизить энергоемкость национального дохода не менее чем в 1,4 раза и металлоемкость почти в 2 раза.

...Увеличить выпуск промышленной продукции не менее чем в 2 раза.

...Обеспечить создание и освоение производства техники новых поколений, позволяющей многократно повысить производительность труда и существенно снизить материальные затраты.

Главная задача двенадцатой пятилетки состоит в повышении темпов и эффективности развития экономики на базе ускорения научно-технического прогресса, технического перевооружения и реконструкции производства, интенсивного использования созданного производственного потенциала, совершенствования системы управления, хозяйственного механизма и в достижении на этой основе дальнейшего подъема благосостояния советского народа.

...Усилить ориентацию научно-технического развития на решение социальных задач.

...Организовать массовый выпуск персональных компьютеров. Обеспечить рост объема производства вычислительной техники в 2—2,3 раза. Высокими темпами наращивать масштабы применения современных высокопроизводительных электронно-вычислительных машин всех классов. Продолжить создание и повысить эффективность работы вычислительных центров коллективного пользования, интегрированных банков данных, сетей обработки и передачи информации.

...Значительно расширить в приборах и средствах автоматизации применение элементной базы повышенной надежности и быстродействия, сверхбольших интегральных схем, лазерной и волоконно-оптической техники.

Из проекта
Основных
направлений
экономического
и социального
развития СССР
на 1986—1990 годы
и на период
до 2000 года

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО НАУКЕ И ТЕХНИКЕ

Издается с 1984 года

ММП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ЧЕТЫРЕ РАЗА В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 4 | 1985 МОСКВА

СОДЕРЖАНИЕ	Ершов А. П.—Колонка редактора	2
МИКРОПРОЦЕССОРНАЯ ТЕХНИКА	Малашевич Б. М., Романов Ф. И.—Перспективная конструкционная система микропроцессорных средств	3
	Кузнецов С. О., Прохоров Н. Л., Раев В. К.—«Электронный диск» на цилиндрических магнитных доменах	11
	Зимовец А. Н., Рыбак К. С., Фомин С. В., Черняк А. Ю., Юрочкин А. Г.—Многомашинная система тестирования микропроцессорных средств вычислительной техники	18
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	Лавров С. С., Слисенко А. О., Цейтин Г. С.—Проект плана-программы по специальности «Информатика и системное программирование»	20
	Бетелин В. Б.—Проблема автоматизации подготовки программной документации	29
	Щелкунов Н. Н., Дианов А. П.—Программирование микросистем реального времени	31
	Лукьянов Д. А.—Как написать кросс-транслятор с языка ассемблер	35
	Липаев В. В., Каганов Ф. А., Керданов А. В., Загубин Ю. В., Колдобский А. Э., Новичков А. С.—Система автоматизации проектирования программ на базе персональных ЭВМ (система ПРА)	42
	Вельбидский И. В., Ковалев А. Л.—Графический стиль программирования для персональной ЭВМ	46
	Лолейт А. Т., Кочегарова С. И.—Опыт применения Р-технологии при разработке программных средств на микроЭВМ	52
ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ	ГИБКИЕ АВТОМАТИЗИРОВАННЫЕ ПРОИЗВОДСТВА	
	Адамов Е. О., Дукарский С. М., Корягин Д. А.—Интегрированная машиностроительная система КАПРИ	53
	Адамов Е. О., Гримайло С. И., Камынин С. С., Корягин Д. А., Смольянов Ю. П.—Роботизированный механообработывающий комплекс для участка гибкого производства	57
	Савин А. И., Кузнецов М. Н., Уксусов А. С., Целяпин А. Н.—Структура системы управления ГАП	62
	Соломенцев Ю. М., Сосонкин В. Л.—Оперативное управление гибкими производственными системами	65
	Каляев А. В., Чернухин Ю. В., Брюхомицкий Ю. А., Каляев И. А., Носков В. П.—Построение однородных управляющих структур адаптивных автономных роботов	68
	Смирнов Ю. С.—Системы управления сервомеханизмами с шаговыми электродвигателями	71
	Бамбуров Н. С., Екимов А. В., Ермолин Ю. В., Мамаков П. В., Сытин А. Н.—Программируемый групповой контроллер на базе МПК БИС серии К580	78
	Петух А. М., Романюк А. Н., Подольский О. А.—Преобразователь интерфейсов ИРПС — «Общая шина»	81
	Михайлов С. А.—Система управления электроприводами судовых грузовых комплексов	83
	Семенов Г. Н.—Автоматизированный комплекс на базе микроЭВМ «Электроника 60М» для управления бурением	86
	Бужан Ю. Д., Осипов И. Н.—Панель управления микропроцессорной системой	87
	Громов Г. Р.—Надежность персональных ЭВМ и производственная загрузка фирменной сети сервиса	89
Рефераты статей	92	
Указатель статей, опубликованных в журнале за 1985 год	95	

Главный редактор

А. П. ЕРШОВ

Редакционная коллегия:

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

Редакционный совет:

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

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

Корректор Т. С. Власкина
 Технический редактор Л. А. Горшкова
 Художник А. П. Бачурин

Адрес редакции: 101820, Москва,
 проезд Серова, 5, редакция журнала
 «Микропроцессорные средства
 и системы».

Телефоны 228-18-88; 221-99-26

Сдано в набор 01.11.85.
 Подписано к печати 02.12.85. Т 21352.
 Формат 84×108^{1/16}. Бумага № 1.
 Высокая печать. Усл. печ. л. 10,08.
 Уч.-изд. л. 15,1. Тираж 21 600.
 Заказ № 376. Цена 1 руб. 10 коп.
 Орган Государственного комитета
 СССР по науке и технике
 Московская типография № 13
 ПО «Периодика» ВО
 «Союзполиграфпром» Государственного
 комитета СССР по делам издательства,
 полиграфии и книжной торговли,
 107005, Москва, Б-5,
 Денисовский пер., дом 30.

На 1-й странице обложки — Неко-
 торые из производственных обла-
 стей применения микропроцессорной
 техники (см. статьи на с. 53, 83, 86)

ОБ ИНФОРМАЦИОННОЙ МОДЕЛИ МАШИНЫ

*Проект директив XXVII съезда КПСС с обязывающей решимостью под-
 черкивает: микропроцессорная вычислительная техника составляет одну из
 основ современного этапа научно-технической революции. Реализация этого
 положения, однако, не придет сама собой вслед за ростом производства вы-
 числительных средств. Мало того, она требует глубокой перестройки всей
 суммы научно-технических знаний инженерного корпуса страны и создания
 новых методов применения этих знаний в практической деятельности.*

*Уже признано, что наиболее массовым применением микропроцессорных
 средств является конструирование и производство машин со встроенными
 микропроцессорами и контроллерами. Однако даже само словопотребление
 в названии этого раздела техники отражает «наружное», вторичное приме-
 нение вычислительных средств: мы сохраняем привычное, сложившееся де-
 сятилетиями видение проектируемой машины и лишь в дополнение к нему
 начинаем соображать, как встроить в машину микропроцессор.*

*По-прежнему главными средствами целостного проектирования машины
 остаются эскиз общего вида, компоновочная схема, весовые, прочностные
 и энергетические показатели. Следующим эшелонем идут кинематические
 схемы, подсистемы электропитания, подачи перерабатываемых или расходуе-
 мых материалов и т. п.*

*Дело теперь состоит в том, что вровень с этими признанными вехами
 создания машины должна стать ее информационная модель. Эта модель име-
 ет дело с теми же частями машины, что и раньше. Однако каждое устрой-
 ство прежде всего рассматривается как источник информационных сигналов
 и как объект применения управляющих воздействий. Информационная модель
 идентифицирует все эти источники информации, определяет их взаимосвя-
 зимость, выражает эту зависимость на языке математических функций или ал-
 горитмов и — самое главное — по такой же схеме включает в общее взаимо-
 действие человека-оператора, по возможности перенося на ЭВМ осуществляе-
 мую им обработку информации.*

*Информационная модель — это не абсолютно новое понятие для маши-
 ностроения. Отдельные ее компоненты применяются уже многие десятилетия
 под названием математических моделей. Прочностный расчет — это особая
 информационная модель взаимодействия материальных частей машины, обме-
 нивающихся информацией в виде значений сил, тензоров напряжений, коэф-
 фициентов упругости, размеров и массы. Другой пример распространенных
 информационных моделей — уравнения автоматического регулирования.*

*Принципиально новым является интеграция информационной модели
 в общий алгоритм функционирования машины и использование не только фи-
 зических зависимостей, но и закономерностей дискретно-логического и ситу-
 ационно-событийного характера. Здесь возникает много новых задач. Нетруд-
 но описать скорость автомобиля в зависимости от его массы, расхода горю-
 чего, уклона дороги и передаточного числа, но гораздо труднее регулировать
 скорость в зависимости от погодных условий или обстановки на дороге.*

*Задача состоит в том, чтобы информационная модель стала таким же
 явным и неотъемлемым компонентом проектирования машины, как и зачаток
 мозга в развитии зародыша, формирующийся и крепнущий вместе с его
 ростом. Понимание необходимости и способность к построению информаци-
 онной модели должны стать такой же глубокой составляющей инженерно-тех-
 нического мышления специалиста, как в свое время сопромат, теоретическая
 механика и электронная схемотехника.*

*Публикуемый в этом выпуске журнала проект так называемого ленин-
 градского учебного плана дает вариант фундаментальной подготовки специа-
 листов, обеспечивающей владение методами построения информационных
 и математических моделей машин и устройств, с которыми наша промыш-
 ленность должна встретит XXI век.*

А. Ершов

УДК 681.3.02

Б. М. Малашевич. Ф. И. Романов

ПЕРСПЕКТИВНАЯ КОНСТРУКЦИОННАЯ СИСТЕМА МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Без четкого и разумного решения вопроса унификации все попытки компьютеризации нашего народного хозяйства обернутся только огромными бесполезными расходами.

Академик А. П. Александров
(«Задача конца столетия», «Известия», 16 октября 1984 г.)

Компьютеризация народного хозяйства на основе традиционных принципов организации разработок и производства средств вычислительной техники, сложившихся в условиях сравнительно малых (сотни и тысячи штук) объемов их выпуска, невозможна. Микропроцессорные средства и системы (МСС) применяются практически во всех областях науки и техники, решают разнообразные прикладные задачи. Попытка широкого внедрения МСС на основе технически и организационно некоординируемых разработок для каждой прикладной задачи неизбежно захлебнется в бесконечной номенклатуре этих систем. Имеется только один путь — комплексная унификация МСС, в том числе унификация конструкций МСС.

До начала «эры микропроцессоров» различные виды оборудования развивались в значительной степени независимо, их межвидовая конструктивная унификация не требовалась и, следовательно, не обеспечивалась. В результате появилось большое число несовместимых конструкций оборудования и применяемой в его составе радиоэлектронной аппаратуры (РЭА), в том числе систем управления. В настоящее время решается задача внедрения МСС во все виды машин, приборов и оборудования (далее — управляемых объектов), т. е. конструктивного встраивания МСС в существующие типы конструкций. Здесь возможны два варианта: адаптация конструкции МСС под различные управляемые объекты или адаптация конструкций управляемых объектов под унифицированную и стандартизованную конструкцию. Последний вариант может показаться нерациональным, так как предполагает перестройку всей промышленности. Но реализация первого варианта не решает проблемы, так как огромное число конструктивных решений МСС резко снизит эффективность их при-

менения: замедлит темпы разработки, производства и внедрения, снизит серийноспособность, а следовательно, и качество, повысит стоимость и т. д.

Оптимальный выбор конструкции МСС может быть сделан только с учетом преобразующего воздействия микропроцессоров на РЭА и управляемые объекты. Оно сравнимо с влиянием введения электропривода в станки и оборудование или двигателя внутреннего сгорания в транспортные средства, коренным образом изменивших принципы построения оборудования. РЭА и управляемые объекты переживают в настоящее время период, аналогичный периоду «самобеглой коляски» в развитии транспортных средств. Действительно, на начальных этапах развития микропроцессорной техники микропроцессорные системы «пристраиваются» к существующему или мало чем отличающемуся от него управляемому объекту (как двигатель внутреннего сгорания устанавливался на шасси автомобиля, напоминающего обычную колесную коляску). Отличие заключается в том, что ранее в оборудование вводился источник механической энергии (двигатель), а теперь вводится интеллект. Следовательно, нужно ожидать еще более резкого, чем это было в автомобилестроении, изменения принципов построения управляемых объектов. В результате чего адаптация их конструкций к унифицированной конструкции МСС может пройти относительно безболезненно. Таким образом, внедрение МСС в народное хозяйство, кроме прямого эффекта — значительного улучшения технических характеристик управляемых объектов, приводит к благоприятному следствию — создает предпосылки для комплексной межвидовой унификации машин, приборов, оборудования. Но это существенно повышает ответственность при выборе принципов по-

строения МСС, в том числе при выборе унифицированных конструкций (УК).

Переоценка роли унификации, попытка ее внедрения там, где она неэффективна, может нанести еще больший ущерб, чем недооценка, так как дискредитирует саму идею. Последующая ее реабилитация — всегда более длительный и болезненный процесс, чем последовательное внедрение.

Рассмотрим возможность и необходимость введения унификации, учитывая все этапы жизненного цикла изделия МСС (разработка, производство, применение) и все связанные с этим стороны (разработчик, изготовитель, потребитель, техническое обслуживание и т. п.). Так, некоторые изделия с технической точки зрения унифицировать необязательно, но эти изделия могут выпускаться предприятием, оснащенным для изготовления унифицированных изделий или их техническое обслуживание возлагается на централизованную службу сервиса. В обоих случаях в целом выпуск унифицированной конструкции может оказаться экономически эффективным.

Унификация конструкций возможна в подавляющем большинстве МСС и РЭА (табл. 1), если КС удовлетворяет следующим основным требованиям: обеспечивает возможность построения изделий с различной степенью стойкости против климатических и механических внешних воздействий и соответствует международным стандартам. Унификация невозможна при встраивании МСС в конструкцию сложной конфигурации управляемого объекта и при построении сверхскоростных МСС.

Пренебрегать последним требованием недопустимо, так как нарушение его приводит к снижению конкурентной способности отечественной продукции на мировом рынке, к за-

труднению эксплуатации импортного оборудования (ремонт, модернизация и т. п.).

Экономические факторы. Области экономической эффективности унификации конструкций РЭА иллюстрируются табл. 2. Рассматриваются три возможных конфигурации прикладных микропроцессорных систем: жесткая, т. е. неизменная за все время жизненного цикла системы; наращиваемая, т. е. допускающая варианты по составу поставки микропроцессорных систем или развиваемая в процессе эксплуатации путем добавления новых устройств; перестраиваемая, т. е. позволяющая потребителю компоновать из имеющегося набора модулей те или иные прикладные системы. Конструкцию систем с жесткой конфигурацией при массовом их производстве унифицировать экономически невыгодно (см. табл. 2). Во всех остальных ситуациях унификация либо эффективна, либо безразлична. Эффект заключается в сокращении сроков разработки, освоения в производстве и внедрения, повышении качества и надежности, упрощении эксплуатации, ремонта, модернизации и т. п.

При оценке экономической целесообразности необходимо учитывать также и структурные особенности РЭА, характер ее составных частей, соответствующих функциональным устройствам. С точки зрения унификации различают устройства трех типов: А — широко применяемые в различных системах; В — применяемые, но не широко; С — уникальные, т. е. применяемые только в данной системе (рассматривается не фактическая, а принципиальная, возможная применяемость). Системы, в которых преимущественно применяются устройства первого типа, всегда экономически эффективнее выполнять на основе унифицированных модулей, второго типа — в зависимости от вида прикладной системы, а применение УК для построения систем, состоящих из устройств третьего типа, требует специального обоснования. В соответствии с вышесказанным (см. табл. 2) устанавливается и область эффективной унификации конструкций МСС.

Таким образом, в большинстве случаев унификация экономически выгодна. Однако при построении прикладных микропроцессорных систем жесткой конфигурации массового производства и МСС, встраиваемых в нестандартные объемы управляемого объекта, необходимо оценить ее последствия в каждом отдельном случае и с учетом всех факторов.

Невозможна в настоящее время полная межвидовая унификация конструкций авиационной аппаратуры с другими видами РЭА, так как для нее проведена международная стандартизация конструкций (на уровне блоков и их объединений), причем

Таблица 1
Области технической возможности унификации конструкций МСС

Особенность управляемого объекта		Технические требования												
		Система размеров	Потребляемая мощность	Отвод тепла	Габариты	Масса	Механическая прочность	Климатическая устойчивость	Требования к влек-тромагниту	Магистрально-модульное построение	Соответствие международным стандартам	Стоимость		
Условия поставки	Промышленность	+	+	+	+	+	+	+	+	+	+	+		
	Культбт	+	+	+	+	+	+	+	+	+	+	М		
	Экспорт	+	+	+	+	+	+	+	+	+	МС	+		
	Импортируемое оборудование	+	+	+	+	+	+	+	+	+	МС	+		
Носимое оборудование		+	М	+	М	М	+	+	+	+	+	+		
Вид исполнения по условиям эксплуатации	Переносное	автомобиле	+	М	+	М	+	Г	Г	+	+	+	+	
		тракторе	+	М	+	М	+	Р	Г	+	+	+	+	
		поезде	+	+	+	+	+	Р	Г	+	+	+	+	
		самолете	+	М	Т	М	М	+	Г	+	+	МС	+	
		корабле	+	+	Т	+	+	+	Г	+	+	+	+	
	Перевозимое		+	+	+	М	+	Р	Г	+	+	+	+	
	Стационарное	В помещении	Вне помещения	+	+	+	+	+	+	Г	+	+	+	+
			не отапливаемом	+	+	+	+	+	+	Г	+	+	+	+
			отапливаемом	+	+	+	+	+	+	+	+	+	+	+
			с кондиционером	+	+	+	+	+	+	+	+	+	+	+
Размещение в объекте	Не встраиваемое		+	+	+	+	+	+	+	+	+	+		
	Встраиваемое	Климатическая зона	без специальных требований	+	+	+	+	+	+	+	+	+	+	
			в объеме заданной формы	-	+	Т	+	+	+	+	+	+	+	
		Север	+	+	+	+	+	+	Г	+	+	+	+	
	Средняя полоса	+	+	+	+	+	+	+	+	+	+	+		
Тропики	+	+	+	+	+	+	Г	+	+	+	+			

Особенность управляемого объекта		Технические требования									
		Система размеров	Потребляемая мощность	Отвод тепла	Габариты	Масса	Механическая прочность	Климатическая устойчивость	Требования к электропитанию	Магнитно-чувствительное построение	Соответствие международным стандартам
Электродинамика	Низко- и среднескоростные	+	+	+	+	+	+	+	+	+	+
	Высокоскоростные	+	+	+	+	+	+	+	+	+	+
	Сверхскоростные	-	+	+	+	+	+	-	+	+	+

Примечание. Р — повышение механической прочности; Г — повышение стойкости против воздействия климатических факторов (герметизация корпуса и т. п.); М — минимизация значения технического параметра; Т — специальные меры для отвода выделяемого тепла; МС — удовлетворение требованиям международных стандартов; + — отсутствие ограничений на унификацию, — — невозможность унификации.

на основе технических решений, явно неприемлемых для других видов РЭА. Однако международные стандарты на авиационную конструкцию периодически совершенствуются (был АРИНК 400, теперь АРИНК 600). В случае широкого внедрения межвидовой унификации конструкций РЭА можно ожидать, что очередной АРИНК будет соответствовать общим решениям. Унификация печатных плат принципиально возможна и в настоящее время, так как они не нормируются стандартами АРИНК.

Во всех остальных случаях (см. табл. 1 и 2) унификация конструкций МСС и других видов РЭА технически возможна и экономически оправдана. Оценим теперь, какой же экономический эффект может принести комплексная унификация МСС и РЭА. При оценке ограничимся рассмотрением эффективности затрат на создание и освоение в производстве МСС. Другие положительные следствия унификации (облегчение автоматизации производства, повышение качества и надежности, сокращение сроков окупаемости, облегчение технического обслуживания и т. д.) также имеют место и достойны специального рассмотрения.

Экономический эффект унификации. Для оценки стоимости разработки и освоения в производстве некоторого числа неунифицированных (N) и унифицированных (M) микропроцессорных систем воспользуемся результатами, полученными расчетным путем и представленными на рис. 1. Затраты на создание 232 унифицированных систем находятся в пределах 2,8...7,6 млн. руб., неунифицированных — 13,5 млн. руб., т. е. экономия от

унификации составляет 5,9...10,7 млн. руб. Или при тех же затратах* может быть создано 529...2250 унифицированных систем. Из рис. 1 также следует, что унификация выгодна при большом числе разрабатываемых систем, эффект ее увеличивается с ростом их числа. При малом числе раз-

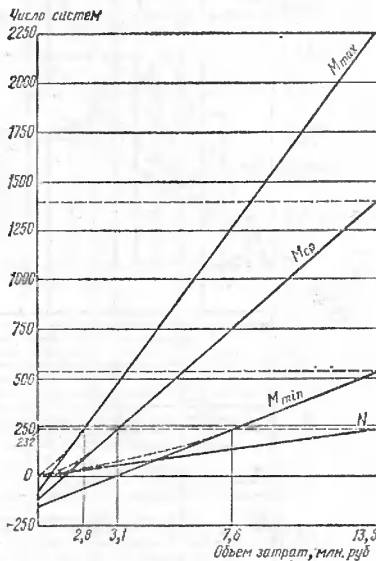


Рис. 1. Оценка экономического эффекта унификации МСС. Зависимость затрат на создание не унифицированных (N) и унифицированных (M) микропроцессорных управляющих систем от числа систем:

M_{max} — максимальная (оптимистическая) оценка; M_{min} — минимальная (пессимистическая оценка); M_{cp} — среднее значение

Области экономической эффективности унификации МСС

Технико-экономические факторы	Конфигурация микропроцессорной системы				
	жесткая	наращиваемая	перестраиваемая		
Разработка	+	+	+		
	Производство	Малосерийное	+	+	+
		Серийное	+	+	+
Применение	Массовое	-	+	+	
	Эксплуатация	0	0	0	
		Ремонт	+	+	+
Модернизация		+	+	+	
Характер составных частей системы	A — широкое применение	+	+	+	
	B — среднее применение	0	+	+	
	C — незначительное применение	-	0	+	
Характер решаемых задач	неизменяемые	-	0	0	
	редко меняющиеся	1	+	+	
	часто меняющиеся	1	+	+	

Примечание. + — унификация выгодна; — — унификация неоправдана; 0 — унификация безразлична; 1 — сочетание не имеет смысла.

рабатываемых систем унификация на основе заранее созданного банка модулей оказывается даже убыточной, так как не оправдываются затраты на создание этого банка. По этой причине при локальной унификации (в рамках предприятия, организации), а она всегда связана с относительно небольшим числом создаваемых систем, банк унифицированных модулей не разрабатывается за-

ранее, а накапливается по мере создания систем. В результате (см. пунктир в начале линий М на рис. 1) экономический эффект от унификации сначала отсутствует, а затем постепенно нелинейно нарастает.

Общая эффективность при локальной унификации обратно пропорциональна числу локальных банков модулей, т. е. прямые М приближаются к прямой N и при бесконечно большом числе банков модулей совпадут с ней. Таким образом, максимальный экономический эффект возможен только при комплексной унификации на основе единого банка стандартных и унифицированных модулей. Необходимо учитывать также, что наиболее полно и быстро указанный эффект может быть получен только при правильной организации внедрения унификации.

Метод внедрения унификации конструкций МСС. В настоящее время сложилась ситуация, когда в стране существует большое число различных конструкций РЭА. Для промышленного выпуска этих конструкций на предприятиях имеется соответствующее, как правило, весьма дорогое и сложное оснащение, отлажены технологические процессы и организовано производство. В этих условиях внедрение новых унифицированных конструкций, даже если все единодушно признают их идеально удовлетворяющими всем требованиям, весьма сложный и болезненный процесс. Внедрение унифицированных конструкций МСС будет обречено на провал, если будет построено на отрицании ранее созданных конструкций, на требовании их немедленной замены. Исторический опыт показывает, что время жизни типовых конструкций РЭА зависит от двух факторов: соответствия их современным техническим требованиям и доступности потребителю. Если конструкция в период ее соответствия техническим требованиям завоевала широкий рынок, она будет жить долго, даже когда перестанет удовлетворять изменившимся требованиям. Так случилось с системой КАМАК, конструкция которой в настоящее время морально устарела, но она широко применяется во всем мире, относительно доступна и не собирается сдавать свои позиции.

Из сказанного следует два вывода. Во-первых, унифицированная КС МСС, если она получит широкое распространение, будет долгоживущей. Во-вторых, внедрять ее нужно через рынок, т. е. дать потребителю на основе крупносерийного и массового производства как базовые конструкции МСС, так и достаточную номенклатуру стандартных и унифицированных микропроцессорных модулей (что возможно, так как эта номенклатура по оценкам специалистов не превышает двухсот наименований). В таких условиях многим изготови-

телям РЭА окажется выгодным не разрабатывать и организовывать производство специализированных составных частей, а покупать их готовыми. В результате большинство существующих конструкций умрет естественной, хотя и ускоренной смертью, а унифицированные КС относительно быстро и безболезненно распространятся в различных отраслях народного хозяйства.

Прежде чем перейти к формированию технических требований к перспективной КС МСС, рассмотрим зарубежный и отечественный опыт построения МСС и РЭА и перспективы развития их конструкций. При этом

примем следующие обозначения размеров: $L \times H \times B$, где L — длина, размер лицевой панели (для печатной платы — толщина основания); H — высота (для печатной платы размер стороны, на которой устанавливаются магистральные соединители); B — ширина (перпендикулярна плоскости лицевой панели).

При анализе будем исходить из того, что основная трудоемкость и материалоемкость при разработке и изготовлении РЭА падает на одно-платные ячейки, а типоразмеры печатных плат обуславливают структуру и принципы построения КС, оптимальность компоновки МСС и т. п. Поэтому при анализе КС в первую очередь будем рассматривать требования к печатным платам.

Анализ зарубежного опыта. Рассмотрим зарубежный опыт построе-

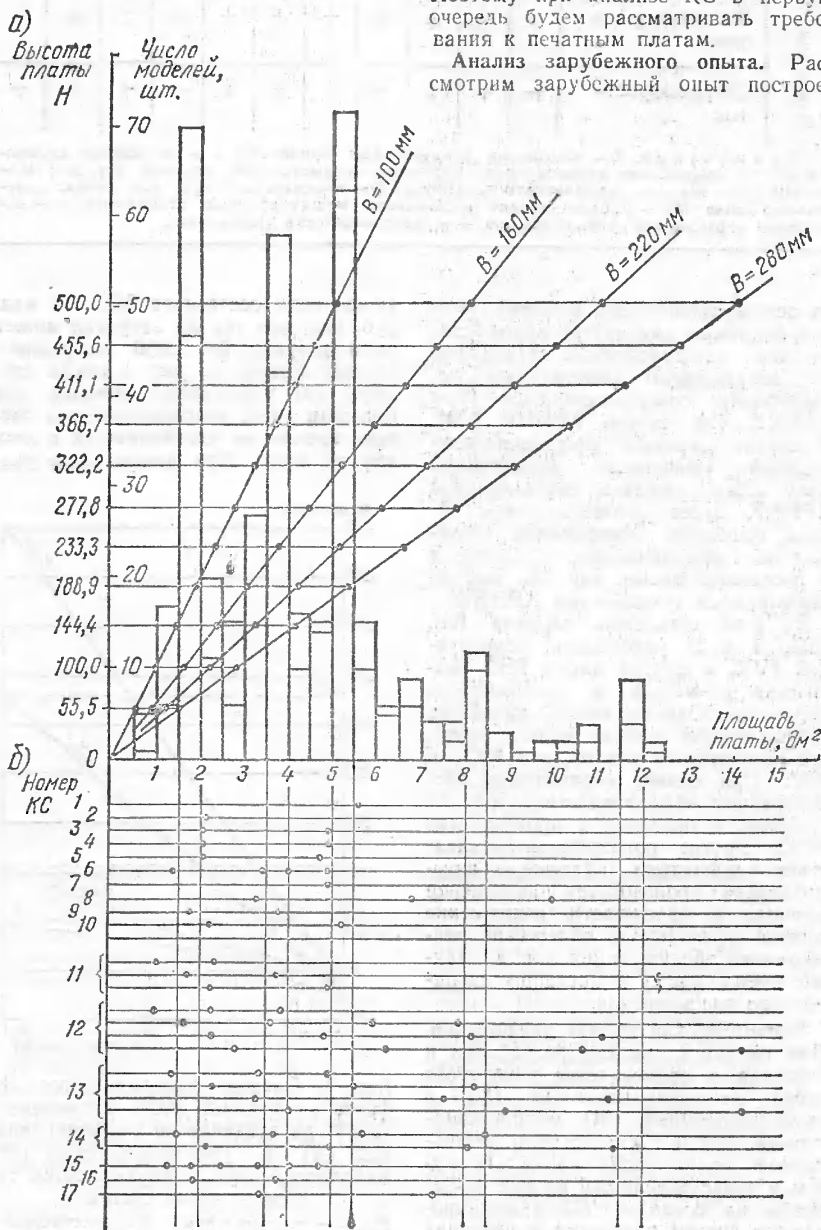


Рис. 2. Распределение зарубежных (а) и отечественных (б) конструктивных систем МСС по площади печатных плат

Основные конструкции РЭА и средств вычислительной техники

№ пп.	Наименование КС (группа КС) Нормативный документ	Обобщенные уровни КС			
		типоразмеры печатной платы, Н×В, мм	частичный корпус	комплектный корпус	комплексный корпус
1	Система КАМАК Крейт и сменные блоки Требования к конструкции и интерфейсу ГОСТ 26.201—80	182,9×305	Сменный модуль	Крейт 5U	
2	Приборы и средства автоматизации ГСП Агрегированные комплексы. Система унифицированных типовых конструкций ГОСТ 20504—75	135×160	Каркас частичный	Каркас блочный	Кожух, стойка, шкаф, тумба, стол
3	Базовые конструкции ЕС ЭВМ ОСТ 4ГО.410.077.4—76	140 } 320 } ×150	ТЭЗ Ячейка	Панель Блок	Стойка, тумба
4	Аппаратура радиоэлектронная. БНК ОСТ 4ГО.410.207—78 ЭВМ	140 } 330 } ×150	Ячейка	Блок	Шкаф, секция, стойка, стеллаж, пульт
5		140 } 305 } ×150			
6	РЭА	170× 75 200 240 280			
7	Корпуса блоков и монтажные устройства пассажирских самолетов ГОСТ 23.701—79 (АРИНК 400)	170×280		Корпус блока	Стеллаж
8	Базовая конструкция микро- и мини-ЭВМ «Электроника»	135 } 280 } ×240 425 }	Каркас	Блок	Стойка
9	Блочные унифицированные конструкции (на основе Европлат). БУК-6 ОСТ 16.0.684.043—78	100,00 } 233,35 } ×160	Каркасы блочные	Каркасы встраиваемые	
10	Базовые конструкции СМ ЭВМ Вторая очередь ОСТ 25.906—80	100,00 } 233,35 } ×220		Каркас блока	Стойка, тумба, кожух, шкаф, стол, подставка

ния МСС и базовых конструкций МСС в двух аспектах: статистику применения печатных плат различной площади и тенденции построения магистрально-модульных систем.

Площади печатных плат. На рис. 2 приведена гистограмма распределения площадей печатных плат 263 реальных одноплатных модулей микропроцессорных средств (косая штриховка) и 120 поставляемых конструктивных систем (без штриховки). Гистограмма имеет пять явно выраженных пиков, распределенных примерно равномерно по шкале площадей печатных плат. Таким образом, статистика подтверждает необходимость вариантности КС по площадям печатных плат для обеспечения оптимальной компоновки функциональных модулей.

В течение ряда лет задачей унификации конструкций занимался специально созданный подкомитет 48Д Международной электротехнической комиссии (МЭК). В настоящее время подкомитет завершает свою работу выпуском международного стандарта МЭК 297, содержащего четыре раздела: МС МЭК 297-1, МС МЭК 297-2, МС МЭК 297-3 и МС МЭК 297-3А и нормализующего все уровни конструктивной иерархии: от печатной платы до стойки. Стандарт предусматривает 44 типоразмера печатных плат (рис. 3), образованных от базовой платы 100×100 мм с помощью модулей приращения ΔВ=60 мм по ширине (четыре значения) и Н=44,45 мм по высоте (одиннадцать значений). Все 44 типоразмера составляют пять модульных по высоте рядов (под модульным понимается ряд типоразмеров плат, высота которых кратна, с учетом зазоров, высоте минимальной платы). На гистограмму (см. рис. 2) наложены все 44 типоразмера печатных плат по МС МЭК 297-3. Наиболее удачно на дике гистограммы накладываются точки, соответствующие печатным платам ряда 2. Это свидетельствует о том, что именно они обеспечивают оптимальную компоновку МСС.

Магистрально-модульные системы. Внедрение микропроцессоров дало новый сильный толчок развитию магистрально-модульных систем [1]. В результате появилось большое число фирменных стандартов на них, начались широкие поставки модулей. Наиболее популярными являются Eurobus (национальный стандарт Великобритании), VME-bus (фирма Motorola Inc., США), EPER (PER Elektronik System, Франция), наборы модулей фирмы Mostek Corp. (США), Philips (США), шины G-64 и G-128 и т. д. Все эти стандарты и модули построены на основе печатных плат ряда 2, получивших наименование Европлат.

Крупнейшие фирмы, например DEC и Intel (США), долгое время исполь-

№ пп.	Наименование КС (группа КС) Нормативный документ	Обобщенные уровни КС			
		типоразмеры печатной платы, Н×В, мм	частичный корпус	комплектный корпус	комплексный корпус
11	Система унифицированных конструкций «Рябина» (на основе Eurobus)	100,00 } 233,35 } × { 100 160 220	Блок частичный	Каркас, кожух	Стол, колонка
12	Единая система стандартов приборостроения (ЕССП) Типовые несущие конструкции ГОСТ 26.204—83	Ряд 3, 6, 9, 12U 100,00 } 233,35 } × { 100 160 220 366,70 } 500,05 } { 280	Кассета, блок	Каркас	Стойка, шкаф
13		Ряд 4, 8, 12U 144,45 } 322,25 } × { 100 160 220 500,05 } { 280			
14	Микропроцессорные средства вычислительной техники и электронное оборудование Конструкционная система	100,00 } 233,35 } × { 100 220 366,70 }	Корпус частичный	Корпус комплектный	Корпус комплексный
15	Базовые несущие конструкции (БНК) РЭА ОСТ 4 ГО 010.223—81	170 × { 75 110 150 200 240 280	БНК-1	БНК-2	БНК-3
16	Проект ГОСТ 605—121—84	170 } 240 } × 150			
17		170 } 360 } × 200			

кое распространение КС, удовлетворяющие этим требованиям, например так называемая Евромеханика, построенная на основе Европлат.

Анализ отечественного опыта [3]. В табл. 3 приведены краткие характеристики 17 наиболее распространенных в стране КС, часть из которых объединена в группы на основе единых нормативных документов. Все эти КС подразделяются на две основные части: варианты (83%) и инвариантные (17%). Как было показано (см. рис. 2) инвариантные КС не обеспечивают возможности оптимальной компоновки МСС и не могут рассматриваться в качестве перспективных. Вариантные КС, в свою очередь, делятся на две группы: на варианты по высоте Н или ширине В.

Конструкции, варианты по высоте. Большинство КС (позиции 3...5, 8...14, 16 и 17 в табл. 3) варианты по высоте Н. Из этих двенадцати КС четыре допускают и варианты ширины печатной платы. Две КС (16 и 17) варианты только по ширине В. Рассмотрим преимущества и недостатки каждой группы КС. Они не имеют принципиальных различий с точки зрения размещения функциональных модулей на печатной плате. Главное требование — число типоразмеров и числовые значения площадей плат, их соответствие ликам гистограммы (см. рис. 2). Но группы КС принципиально не равноценны при компоновке блоков. Печатные платы различной высоты при условии модульности размеров Н оптимально компонуются в блоке (рис. 3 и 4). Такое построение обеспечивает конструктивную гибкость, т. е. дает возможность добавлять модули в резервные посадочные места блока (наращивание системы), заменять морально устаревшие модули новыми, меньшего размера, с высвобождением резерва для наращивания, либо того же размера, но с большими функциональными возможностями (модернизация системы), извлекать все старые модули и компоновать новую систему из старых и новых модулей (реконструкция системы). При этом объем блока используется максимально, а межплатные электрические связи предельно упрощаются, так как выполняются в одной плоскости по задней стороне блока. При немодульных размерах высоты плат ухудшается использование объема блока, но возможно сохранение удобства межплатных соединений.

Конструкции, варианты по ширине. При компоновке плат различных по ширине, объем блока используется неэффективно (потери — поз. 1 на рис. 5), а межплатные электрические связи крайне затруднены, так как выполняются в нескольких плоскостях (на рис. 5 приведен пример компоновки КС 16 по табл. 3 — шесть плоскостей). Некоторое улуч-

зовали свои уникальные фирменные стандарты, в том числе на конструкции, в качестве средства закабаления потребителя, однажды купившего их продукцию [2]. Практически всегда потребитель наращивает однажды купленные средства вычислительной техники, т. е. покупает дополнительные модули и новые системы. Уникальность фирменных стандартов обрекает его на покупку изделий одной фирмы, так как изделия других фирм несовместимы с имеющимися у него.

Однако за последние годы большое число мелких фирм, в основном в Европе, а также в США, объединились вокруг МС МЭК 297 и в сово-

купности создали мощную конкуренцию крупным фирмам. В результате фирма DEC вынуждена была выпустить микроЭВМ МХЕ 11 (конструктивный вариант LSI-11/23) и ряд модулей к ней на Европлатах. Фирма Intel Inc. предложила проект нового, в настоящее время самого популярного стандарта на магистрально-модульную систему Multibus II, основанного также на Европлатах.

Таким образом, КС должны быть вариантами по площадям печатных плат и должен быть статистически сформирован оптимальный ряд величин этих площадей. За рубежом имеются и получают все более широ-

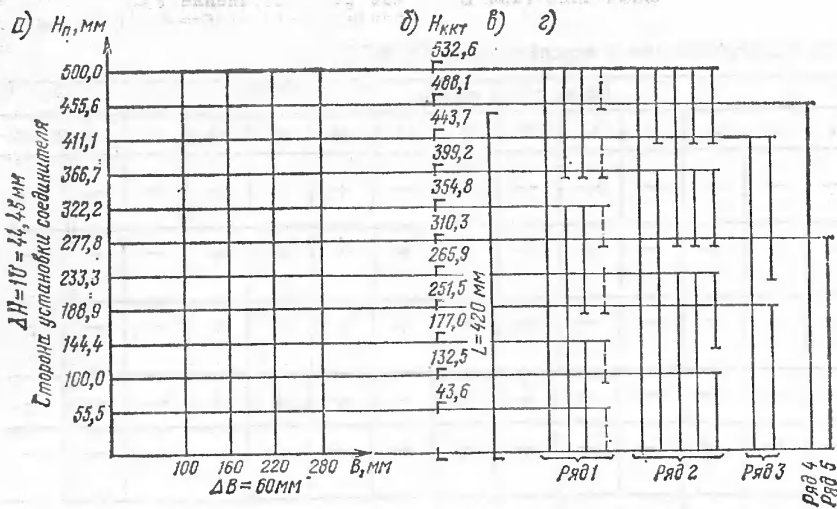


Рис. 3. Печатные платы по ГОСТ 26.204-83:

а — номенклатура типоразмеров; б — размеры высоты H комплектных корпусов, соответствующих печатным платам; в — длина проема комплектного корпуса; г — модульные ряды печатных плат и их компоновка

шение использования объема блока возможно при помощи составных, двухплатных ячеек (поз. 2), однако при этом нарушается магистрально-модульный принцип построения системы, так как удаленная от задней стенки блока плата (нижняя поз. 2) не имеет возможности соединения с межплатной магистралью, следовательно, использование составных плат ограничено.

Таким образом, печатные платы, варианты по высоте, имеют принципиальные преимущества. Однако иногда, при встраивании микропроцессорных систем в управляемый объект сложной конфигурации (см. табл. 1) бывают необходимы печатные платы различные именно по ширине. В таких случаях на основе функционально-стоимостного анализа следует сделать выбор между унифицированной или специализированной конструкциями.

Распределение печатных плат всех рассматриваемых КС по гистограмме площадей плат приведено на рис. 2, причем в КС, вариантных по высоте и ширине печатных плат, точки, отражающие площадь плат с каждым значением ширины B , проставлены вдоль отдельной линии. Распределение показывает, что площади печатных плат КС в основном довольно удачно совпадают (или близки) с пиками гистограммы. Однако большинство КС не располагают всем спектром оптимальных площадей печатных плат.

Особое внимание следует обратить на КС 6 и КС 15. Из рис. 2 видно, что в этих КС неоправданно большая номенклатура печатных плат с близкими значениями площадей и далеко не полный спектр оптимальных площадей печатных плат. Это объясняется тем, что указанные КС яв-

ляются искусственным объединением печатных плат, имеющих одинаковую высоту $H=170$ мм и редко применяемых совместно. Кроме того, реальные конструкции одноплатных ячеек, построенных на печатных платах КС 6 и КС 15, выполняются с различными соединителями (СНП 34, КРПМ, РППМ 27, ОНП-ВГ-35, СНП 49 и др.), различными лицевыми панелями или рычагами, т. е. практически не совместимы друг с другом.

Полный спектр площадей печатных плат с минимальной избыточностью содержит только КС 12 ($B=160, 220$), выполненная в соответствии с ГОСТ 26.204—83. КС 12 и совместимые с ней КС 9...11 в настоящее время наиболее распространены в нашей стране (их применяют предприятия не менее восьми министерств и ведомств) и странах СЭВ (это вариант вышеуказанной Евромеханики).

В настоящее время и на ближайшее обозримое будущее основным при построении МСС и многих других видов РЭА является магистрально-модульный принцип [1]. Следовательно, КС МСС должна обеспечивать возможность построения различных по сложности функциональных модулей при оптимальной плотности компоновки элементов. В связи с этим уместно остановиться на бытующем мнении, что рост степени интеграции микросхем приводит к необходимости увеличения площади печатных плат и числа контактов межплатных соединений. И то и другое существенно усложняет конструкцию МСС. Это утверждение действительно справедливо, но только в тех случаях, когда не применяется магистрально-модульный принцип построения, т. е. когда расчленение функциональных устройств на функ-

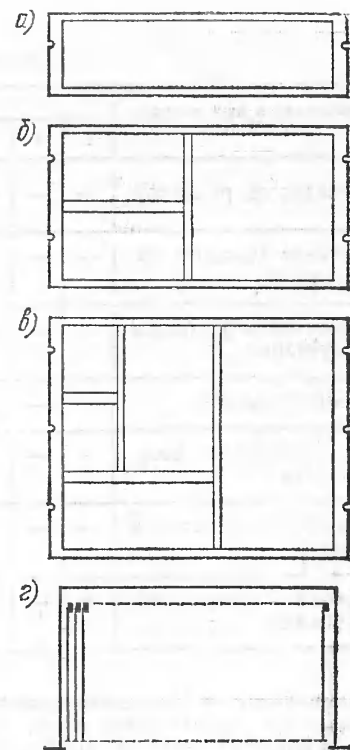


Рис. 4. Компоновка вариантных по высоте печатных плат в комплектном корпусе а, б, в — вид спереди:

а — из плат одного типоразмера; б — из модульных плат двух типоразмеров; в — из модульных плат трех типоразмеров. г — вид сверху

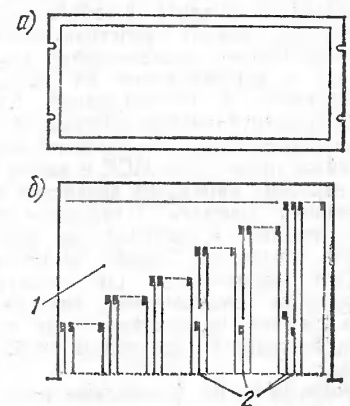


Рис. 5. Компоновка вариантных по ширине печатных плат в комплектном корпусе:

а — вид спереди; б — вид сверху (1 — высота корпуса, 2 — составные печатные платы)

Соответствие КС требованиям к перспективной КС МСС

Требования к перспективной КС МСС	Номер КС по табл. 3																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Вариантность размеров	-	-	-	-	-	-	-	-	-	-	-	+	+	+	-	-	-
Сочетание конструктивных единиц	-	-	-	-	+	-	-	+	+	+	+	+	+	+	-	-	-
Вариантность условиям эксплуатации	-	-	-	-	-	-	-	-	+	+	+	+	+	+	-	-	+
Соответствие МС	-	-	-	-	-	-	-	-	+	+	+	+	+	+	-	-	-
Технологическая независимость	+	-	-	-	-	-	-	-	+	+	+	+	+	+	-	-	-
Широкое применение в стране	+	-	-	-	-	-	-	-	+	+	+	+	-	+	-	-	-
Широкое применение за рубежом	+	-	-	-	-	-	-	-	-	+	+	+	-	+	-	-	-

ствиями являются сложность проверки, наладки, ремонта таких ячеек, так как каждая из них не выполняет какой-либо полной функции. Преимуществом такого подхода (небесспорным) является возможность обеспечения максимальной плотности компоновки МСС. Применение магистрально-модульного принципа построения МСС требует вариантов по площади печатных плат (см. высокие пики гистограммы на рис. 2). Применение стандартных интерфейсов не увеличивает, а уменьшает число межплатных связей. Таким образом, магистрально-модульный принцип построения МСС наряду со своими прямыми преимуществами еще и способствует решению проблемы конструирования МСС. На основании проведенного анализа и обобщения на других уровнях конструктивной иерархии можно сформулировать требования к перспективной КС МСС.

Требования к перспективной КС МСС. Конструкционная система, которая может обеспечить наиболее эффективное применение МСС в различных отраслях народного хозяйства и гармонично сочетать технические и экономические требования на всех этапах жизненного цикла изделия МСС и управляемого им объекта (разработка, производство, эксплуатация, ремонт, модернизация) должна удовлетворять следующим требованиям:

вариантность по физическим параметрам (площадям печатных плат, объемам частичных, комплектных и комплексных корпусов);

гармоничное сочетание конструктивных единиц с разными физическими параметрами в конструкции изделия без дополнительных наклад-

ных расходов (потери объема, сложность механического и электрического соединений и т. п.);

вариантность и адаптируемость к различным условиям эксплуатации, т. е. повышение стойкости против механических и климатических воздействий;

соответствие международным стандартам в пределах, обеспечивающих совместимость и взаимозаменяемость функционально-конструктивных модулей;

технологическая независимость, обеспечивающая автоматическую совместимость модулей, изготовленных по разным технологиям;

широкое применение КС в стране и за рубежом.

Выбор перспективной КС МСС. При выборе перспективной КС МСС возможны два пути: создание новой КС или выбор из числа существующих (возможно, с доработкой или развитием). Требование соответствия международным стандартам ограничивает возможность создания новой КС, так как стандарты и соответствующая им КС существует и нецелесообразно их разрабатывать вновь. Однако нужно учитывать, что технический комитет 48 МЭК недавно образовал рабочую группу для разработки новой, чисто метрической КС, завершение работы группы (на основании опыта работы ПК 48Д МЭК) можно ожидать через 10...20 лет.

В табл. 4 приведено соответствие существующих КС приведенным выше техническим требованиям. Рассмотрение таблицы показывает, что всем требованиям удовлетворяет только КС 12, т. е. Евромеханика, расширенная платами 9U и 12U. Та-

ким образом, проведенный анализ показал, что в качестве перспективной КС МСС может быть рекомендована только Евромеханика, соответствующая требованиям МС 297 МЭК (СТ СЭВ 3266—81, ГОСТ 26.204—83). Правильность этого вывода подтверждается широким ее применением как в нашей стране, так и за рубежом.

ЛИТЕРАТУРА

1. Малашевич Б. М., Дшхунян В. Л., Борщенко Ю. И., Никольский О. А., Хорин В. С. Магистрально-модульные микропроцессорные системы. — Микропроцессорные средства и системы, 1984, № 4, с. 3—11.
2. Киреев А. Н. Стандартизация в США: экономические аспекты и проблемы организации. — М.: Стандарты, 1982.
3. Романов Ф. И., Шахнов В. А. Конструкционные системы микроЭВМ. — М.: Радио и связь, 1983.

Статья поступила 12 июня 1985 г.

РЖ ВИНТИ, АВТ-85

8.81.359. Робот с управлением от ЭВМ. 1984, (англ.)

Для демонстрации техники сопряжения робота с микроЭВМ рассматриваются особенности сопряжения робота типа Minimover-5 с микроЭВМ типа Vic-20. Сопряжение обеспечивается как со стороны плеча робота, так и со стороны ЭВМ через 8-рядную параллельную шину, которая в ЭВМ является шиной ввода-вывода пользователя. Связь ЭВМ с роботом выполняется с помощью кабеля через 24-контактный разъем, установленный на ЭВМ, и через 40-контактный разъем, установленный на блоке связи (входе).

«ЭЛЕКТРОННЫЙ ДИСК» НА ЦИЛИНДРИЧЕСКИХ МАГНИТНЫХ ДОМЕНАХ

Электромеханические устройства внешней памяти вращающегося типа — часто одно из слабых звеньев микропроцессорных систем. Заметно уступающая электронная память во времени наработки на отказ, энергопотреблении, габаритах, массе, температурной стойкости и способности работать в загрязненных и запыленных средах, они часто тормозят создание новых надежных систем управления и контроля, особенно во встраиваемом и портативном исполнении.

Использование накопителей на гибких магнитных дисках (НГМД), в которых поверхность диска постоянно находится в контакте с головкой записи-считывания, изготовителя обуславливают рядом жестких требований [1], далеко не всегда выполняемых на практике. Это неизбежно уменьшает время наработки на отказ,

Поэтому целесообразна разработка и промышленное освоение бездисковых систем внешней памяти, основанных на новых физических принципах и твердотельной технологии. Системы внешней памяти на цилиндрических магнитных доменах (ЦМД) ЗУ [2—4] «умеют делать» лучше все, на что способны магнитные диски, но пока уступают мощным дисковым системам в абсолютных объемах памяти и относительной цене бита хранимой информации.

На диаграмме (рис. 1) ЗУ на ЦМД представлены двумя типами: одноплатными (накопитель и контроллер на одной плате) и многоплатными, в которых один контроллер обслуживает 8...16 плат (модулей) накопителя. При этом на каждой плате размещают не менее восьми ЦМД-микросхем. Как следует из диаграммы, уже существующие средства памяти на ЦМД способ-

ны во многом удовлетворить потребности малых и микроЭВМ во внешних ЗУ типа НГМД. Заметим, что попытки микроминиатюризации НМД и НГМД, как правило, влекут за собой уменьшение времени наработки на отказ и увеличение удельной цены. Поэтому именно в сфере применений НГМД и микромини-НМД ситуация особенно благоприятна для внедрения электронной памяти на ЦМД, улучшение технико-экономических показателей которой прямо связано с массовостью выпуска. Уже сейчас, когда промышленностью освоены ЦМД-микросхемы емкостью 256 Кбит...4 Мбит, разработчики систем внешней памяти имеют возможность создавать ЦМД-ЗУ, во многом превосходящие по своим показателям электромеханические [1—6].

Из-за отсутствия операционных систем, позволяющих обслуживать

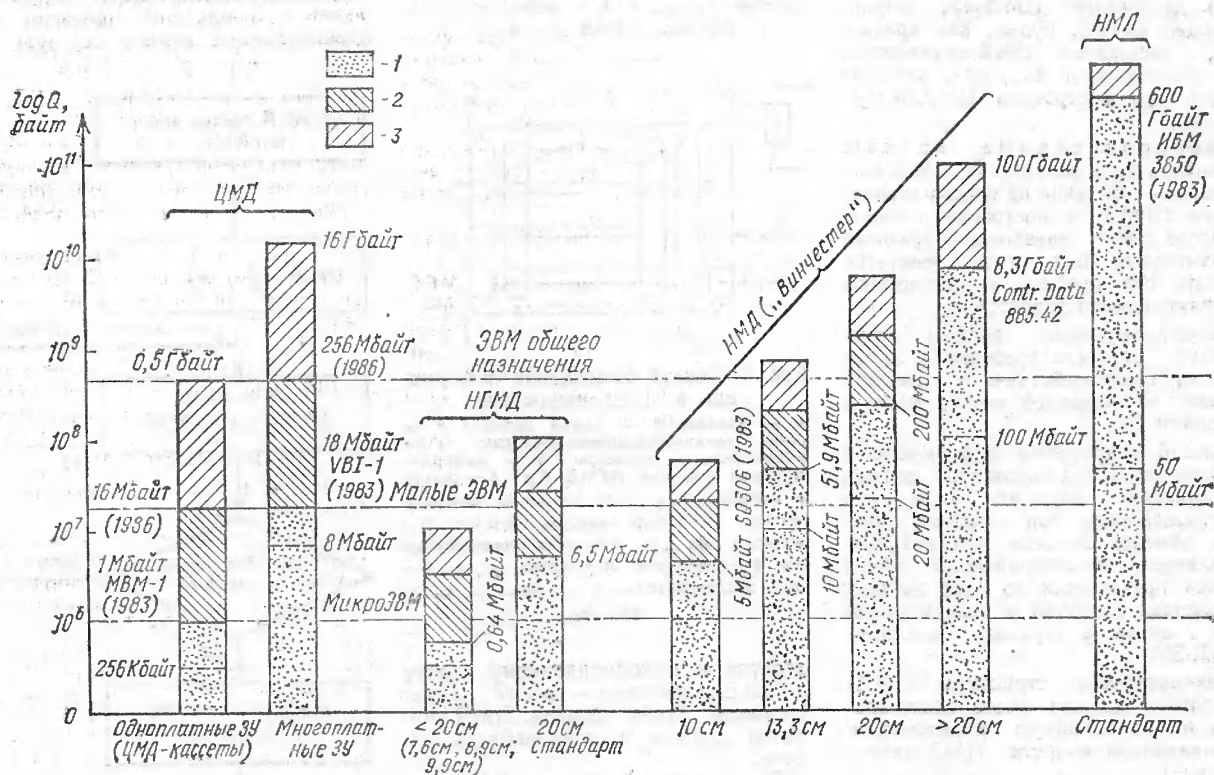


Рис. 1. Сравнение различных типов внешних ЗУ:

1 — промышленный выпуск, 2 — опытно-конструкторские разработки и год начала промышленного выпуска, 3 — научный задел с выходом на промышленный выпуск в 1995—2000 гг.

ЦМД-ЗУ, одним из эффективных путей их внедрения на «рынок» внешней памяти на данном этапе следует считать эмуляцию (имитацию) поддерживаемых существующими операционными системами внешних ЗУ (прежде всего, НГМД). При этом ЦМД-ЗУ должно быть построено так, чтобы система «не чувствовала разницы» между эмулятором и имитируемым ВЗУ, т. е. должна использовать протокол обмена последнего при взаимодействиях с центральным процессором системы [6, 7]. Такой подход к созданию ЦМД-памяти позволит пользователю без каких-либо изменений в программном обеспечении замеснить ненадежные электромеханические устройства на их магнитоэлектронные аналоги. Этот подход будет, видимо, доминирующим для обновления внешних ЗУ уже существующих систем. Что касается перспективных разработок вычислительных средств, то для перехода к бездисковым системам возможны: создание математического обеспечения (позволяющего в полной мере использовать достоинства ЦМД-ЗУ) и разработка ЦМД-накопителей (максимально использующих уникальные свойства ЦМД-сред, например, возможное создание иерархических структур памяти при комбинации логических и запоминающих массивов в одном кристалле и др.).

В дальнейшем ЦМД-ЗУ, эмулирующее НГМД, будем, для краткости, называть ЦМД-эмулятором. Рассмотрим круг вопросов, возникающих при построении ЦМД-эмуляторов.

Первоочередные задачи. Основная цель разработки ЦМД-эмуляторов — создание не просто эквивалента НГМД, а построение электронного ЗУ с заведомо лучшими параметрами. Для этого требуется решить ряд проблем теоретического и практического характера:

способ эмуляции должен оптимально учитывать требования системы по быстродействию и способу обмена информацией между ЭВМ и внешним ЗУ;

способ размещения информации в накопителе ЦМД-эмулятора должен оптимально улучшать параметры быстродействия при многостраничном обмене. Отметим, что в ЦМД-эмуляторах многостраничный обмен нужен практически во всех случаях вследствие различий в длинах секторов диска и страниц массивов ЦМД-ЗУ;

накопительная структура ЦМД-эмулятора должна иметь минимальную информационную избыточность при заданной емкости ЦМД-микросхемы;

конструктивное исполнение ЦМД-эмулятора должно допускать простую смену носителя.

К выбору способа эмуляции

Поскольку каждый тип внешних устройств характеризуется своим особым набором действий, совершаемых при вводе-выводе, программа обмена данными строится специально для каждого типа внешних устройств. Поэтому основная задача ЦМД-эмуляции — имитация действий НГМД при операциях ввода-вывода. Для доступа к информационному блоку (сектору) на НГМД (рис. 2, 3) необходимо задать адрес дорожки и адрес сектора, а для

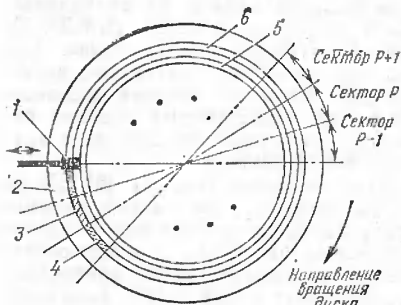


Рис. 2. Способ размещения информации на магнитном диске:

1 — головка «запись-считывание»; 2 — гибкий магнитный диск; 3 — сектор с адресом $A_{i,j}$, где i — адрес дорожки диска, j — номер сектора; 4 — сектор с адресом $A_{i,j+1}$; 5, 6 — информационные дорожки НГМД A_i, A_{i+1}

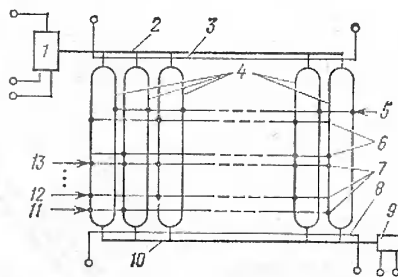


Рис. 3. Способ размещения информации в ЦМД-микросхеме:

1 — датчик; 2 — канал вывода; 3 — шина репликатора/переключателя; 4 — накопительные регистры; 5 — информационная дорожка НГМД A_i ; 6 — сектор с адресом $A_{i,j}$, где i — адрес дорожки диска, j — номер сектора; 7 — сектор с адресом $A_{i,j+1}$; 8 — шина переключателя; 9 — генератор; 10 — канал ввода; 11—13 — последовательные страницы сектора $A_{i,j}$

доступа к информационному блоку в ЦМД-накопителе — только адрес страницы. Таким образом, ЦМД-эмулятор должен преобразовывать адреса.

По способу преобразования адреса ЦМД-эмуляторы разделим на два класса: с внутренней и с внешней эмуляцией.

При внутренней эмуляции адреса обращения преобразует контроллер ЦМД-эмулятора. При этом значение адреса страницы из адреса обращения к диску может быть получено одним из трех способов: программным, аппаратным и аппаратно-программным.

При программном преобразовании адреса контроллер ЦМД-эмулятора вычисляет $A_{ЦМД}$ (рис. 4, табл. 1). Длина подпрограммы преобразования — 57 байт, а максимальное время выполнения — 647 мкс. Реальное время выполнения программы на практике существенно меньше и зависит от конкретных значений параметров k_1 и k_2 .

Для аппаратного преобразования адреса в контроллере необходима дополнительная аппаратура, например декодер адреса, построенный на базе стандартных микросхем ПЗУ. На адресные входы декодера поступает адрес обращения в виде номера дорожки и номера сектора, а с выхода снимается адрес страницы обращения ЦМД-эмулятора (рис. 5, а). Наиболее распространенные форматы НГМД приведены в табл. 2.

Наибольшая емкость ПЗУ потребуются при эмуляции НГМД, имеющего 77 дорожек по 26 секторов в каждой (табл. 2). Для адресации такого НГМД необходим 13-разрядный регистр адреса (7 старших разрядов для идентификации номера дорожки и 5 младших разрядов для идентификации номера сектора), а

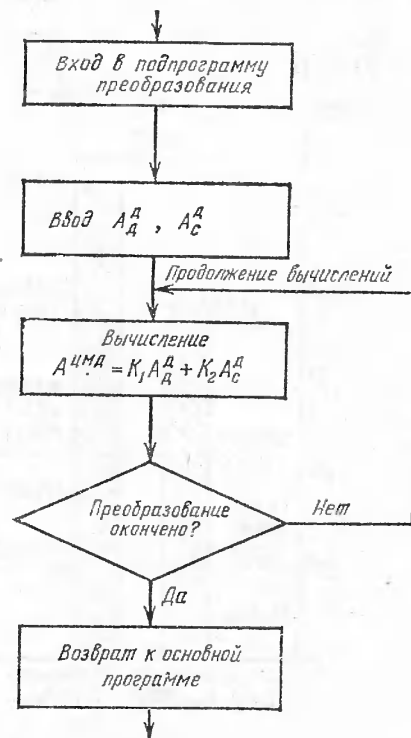


Рис. 4. Блок-схема алгоритма программы преобразования адреса

Программа преобразования дискового адреса в адрес страницы ЦМД-накопителя для БИС КР5801К80А

Метка	Код команды	Операнд	Комментарий	Число байтов	Время выполнения, мкс
1	2	3	4	5	6
BUBBL	LHLD		Загрузка A_d^D и A_c^D в регистры H и L	3	8
	MOV	B, L	$(L) \rightarrow B (A_d^D \rightarrow B)$	1	2,5
	MVI	E, K1	Загрузка коэффициента k_1 в регистр E	2	3,5
	MVI	L, 2	Инициализация счетчика байтов	2	3,5
M0	MVI	D, 0	Сброс старшего байта произведения	2	3,5
	MVI	C, 8	Инициализация счетчика битов	2	3,5
M1	MOV	A, E	Множитель — в аккумулятор	3	2
	RAR		Очередной бит во флажке C	3	2
	MOV	E, A	Возвращение множителя	1	2,5
	DCR	C	Декремент счетчика битов	1	2,5
	JZ	M3	Умножение закончено?	1	2,5
	MOV	A, D	Старший байт произведения	3	5
	JNC	M2	Бит множителя равен нулю?	1	2,5
	ADD	B	Суммирование множителя	3	5
M2	RAR		Сдвиг частичной суммы	3	2
	MOV	D, A	Возвращение старшего байта	3	2
	JMP	M1	Умножение на следующий бит	1	2,5
M3	DCR	L	Декремент счетчика байтов	3	5
	JZ	M4	Выход из большого цикла	1	2,5
	XCHG		Обмен содержимым регистров	3	5
	SHLD	$\langle B_2 \rangle, \langle B_3 \rangle$	$A_d^D \cdot k_1 \rightarrow M$	1	2
	XCHG		Обмен содержимым регистров	3	8
	MVI	B, H	$(H) \rightarrow B (A_c^D \rightarrow B)$	1	2
	MVI	E, K2	$k_2 \rightarrow E$	2	3,5
	JMP	M0	Умножение $A_c^D \cdot k_2$	2	3,5
M4	LHLD		Загрузка $A_d^D \cdot k_2$	3	5
	DAD	D	в регистры Сложение регистров H, L, D, E	3	8
	SHLD		$A_{стр}^{ЦМД} = k_1 A_d^D + k_2 A_c^D$	1	5
	RET		Возврат к основной программе	3	8
				1	5,5

декодирующее ПЗУ должно иметь, соответственно, организацию $8 \text{ Кбит} \times 13$. Для построения ПЗУ с данной организацией целесообразно использовать две микросхемы МПЗУ с организацией $8 \text{ Кбит} \times 8$.

При аппаратно-программном преобразовании адреса контроллер вычисляет адрес обращения, используя результаты промежуточных операций. Этот способ уменьшает время преобразования адреса (по сравнению с чисто программным) до 36 мкс (табл. 3) и уменьшает емкость декодирующего ПЗУ по сравнению с аппаратным способом преобразования до 512×16 бит. В этом случае в качестве декодирующего ПЗУ применимы микросхемы ППЗУ.

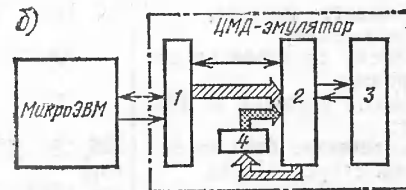
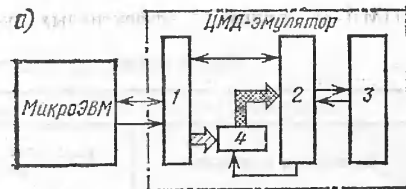


Рис. 5. Упрощенные структурные схемы ЦМД-эмуляторов с аппаратным (а) и аппаратно-программным (б) способами преобразования адреса: 1 — блок интерфейса; 2 — контроллер эмулятора; 3 — накопитель; 4 — блок (частичный для случая б) преобразования адреса

При внешней эмуляции система памяти на ЦМД подключается к ЭВМ через контроллер ввода-вывода (адаптер). Помимо преобразования адреса адаптер «стыкует» интерфейсы (рис. 6). Преимущества такого подхода к созданию систем памяти на ЦМД по сравнению с

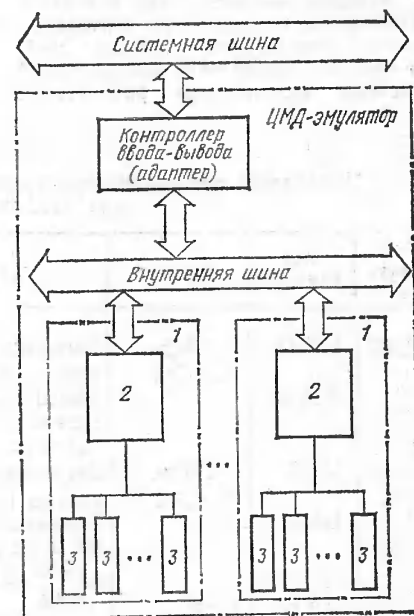


Рис. 6. Структурная схема ЦМД-ЭУ с внешней эмуляцией: 1 — модуль ЗУ; 2 — контроллер модуля; 3 — накопитель

Таблица 2

Сопоставление различных форматов НГМД

Организация данных и емкость — стандартные 8-дюймовые (20 см) форматы						
Одinarная плотность	Наиболее употребимые форматы			Прочие форматы		
Емкость сектора, байт/сектор	128	256	512	1024	2048	4096
Число секторов на дорожке	26	15	8	4	2	1
Число дорожек на диске	77	77	77	256 адресуемых		
Количество байт на одной стороне диска	256, 256 (128 байт/сектор) 295, 680 (256 байт/сектор) 315, 392 (512 байт/сектор)			315, 392 (77 дорожек)		
Двойная плотность						
Емкость сектора	256	512	1024	2048	4096	8192
Число секторов на дорожке	26	15	8	4	2	1
Число дорожек на диске	77	77	77	256 адресуемых		
Количество байт на одной стороне диска	512, 512 (256 байт/сектор) 591, 360 (512 байт/сектор) 784 (1024 байт/сектор)			630, 784 (77 дорожек)		

внутренней эмуляцией: 1) возможность использования ЦМД-ЗУ как в функции эмулятора, так и в самостоятельной функции при непосредственном подключении к шинам ЭВМ и наличии специальной операционной системы, позволяющей работать с

ЦМД-ЗУ; 2) к одному адаптеру можно подключать несколько ЦМД-ЗУ для увеличения ресурса памяти при минимальных затратах на средства эмуляции; 3) гибкая перестройка на эмуляцию НГМД другого типа, смена адаптера (или

Таблица 3

Программа аппаратно-программного преобразования адреса для БИС КР580ИК80А

Метка	Код команды	Операнд	Комментарий	Число байтов	Время выполнения, мкс
BUB2	LHLD	[$\langle B_2 \rangle$, $\langle B_3 \rangle$]	Загрузка $A_{д}^D \cdot k_1$ в регистры H, L	3	8
	XCHG		Обмен содержимых регистров (H) \rightarrow (D); (L) \rightarrow (E)	1	2
	LHLD	[$\langle B_2 \rangle$, $\langle B_3 \rangle$]	Загрузка $A_{с}^D \cdot k_2$ в регистры H, L	3	8
	DAD	D	Сложение регистров (H, L) с (D, E) ($A_{ЦМД} \rightarrow H, L$)	1	5
	SHLD	[$\langle B_2 \rangle$, $\langle B_3 \rangle$]	$A_{ЦМД}$ в память	3	8
	RET	ET	Возврат к основной программе	1	5,5

его перепрограммирование) без изменения структуры ЦМД-ЗУ. Для адаптера преобразование адреса можно реализовать так же, как и для внутренней эмуляции. Отметим, что при использовании ЦМД-ЗУ в «самостоятельной роли», т. е. при обслуживании его программой-драйвером, последнюю можно построить так, что ЭВМ и ЦМД-ЗУ будут обмениваться информационными блоками, эквивалентными сектору НГМД [7]. В этом случае адреса преобразует центральный процессор. Однако подобный подход к эмуляции неизбежно снижает общую производительность системы из-за роста дополнительного времени на преадресацию. Это, в свою очередь, ограничит область применения таких драйверов, например для устройств с прямым доступом к памяти.

В табл. 4 сведены характерные особенности рассмотренных выше способов эмуляции. Отметим, что способы эмуляции в каждом конкретном случае необходимо выбирать с учетом структуры накопителя ЦМД-эмулятора и способа размещения информации в последнем. Ниже обсуждаются некоторые целесообразные способы размещения информации в ЦМД-эмуляторе.

К выбору способа размещения информации в накопителе ЦМД-эмулятора

Параметры одной из ЦМД-микросхем (К1605РЦ1 [11]) емкостью 256 Кбит приведены ниже:

Номинальная емкость, Кбит	256
Полная емкость, бит	289 050
Частота управляющего поля, кГц	100—150
Емкость одного НР, бит	1025
Общее количество НР, шт.	282
Количество годных НР, шт.	260

Размер информационной страницы, определяемый числом накопительных регистров ЦМД-микросхемы, как правило, меньше размера секторов (см. табл. 2), которыми обмениваются ЭВМ и НГМД. Поэтому при эмуляции НГМД информационный сектор будет занимать несколько информационных страниц. В зависимости от организации ЦМД-накопителя информационный сектор может быть размещен в ЦМД-микросхемах одним из трех способов (рис. 7): а) одностраничным, б) кратностраничным и в) разноформатным.

При одностраничном размещении информации в каждой ЦМД-микросхеме, участвующей в обмене, размещается по одной странице каждого информационного сектора (рис. 7, а).

При кратностраничном размещении информации в каждой ЦМД-микросхеме размещается l (l —целое

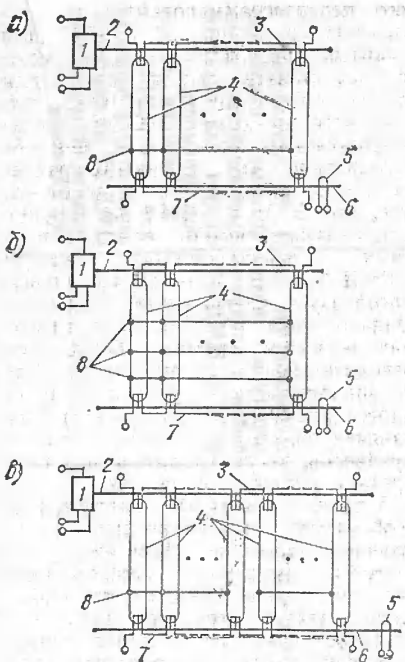


Рис. 7. Способы размещения информации в ЦМД-накопителе:

а) одностраничный, б) кратностраничный, в) разноформатный; 1 — датчик; 2 — канал ввода; 3 — канал вывода; 4 — накопительные регистры; 5 — генератор; 6 — канал ввода; 7 — переключатель; 8 — репликатор/переклю­чат­ель; 8 — информационный сектор с адресом A_{1j}

число) страниц одного информационного сектора (рис. 7, б).

Случай, когда l — не целое число, (рис. 7, в) соответствует разноформатному способу размещения информации, применять который в за-

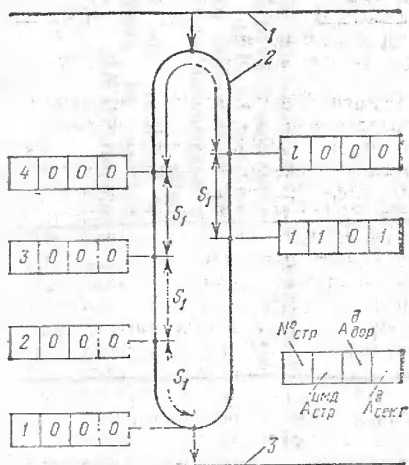


Рис. 8. Равноинтервальная синхронизация адресов страниц:

1 — канал ввода; 2 — накопительный регистр; 3 — канал вывода; A — адрес первой страницы сектора; $N_{стр}$ — номер страницы в секторе

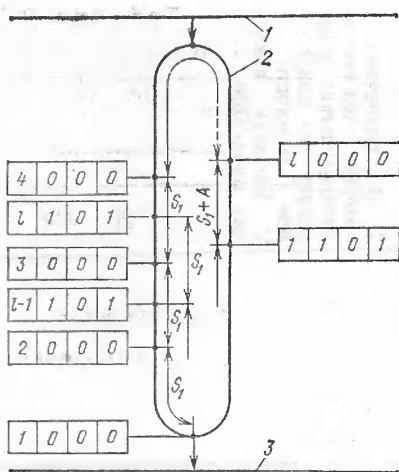


Рис. 9. Модифицированная равноинтервальная синхронизация:

1 — канал ввода; 2 — накопительный регистр; 3 — канал вывода

дачах эмуляции нецелесообразно ввиду неиспользования до 50% информационной емкости ЦМД-микросхем при работе с незаполненными страницами. Заполнение же страниц сопровождается увеличением среднего времени доступа к сектору и увеличением цикла обращения [9].

Итак, одностраничный и кратностраничный способы размещения информации, вероятно, наиболее приемлемы в ЦМД-эмуляторах. Эффективность использования того или иного способа размещения информации во многом определится способами синхронизации физических и логических адресов страниц (рис. 8... 10) и характером обращения.

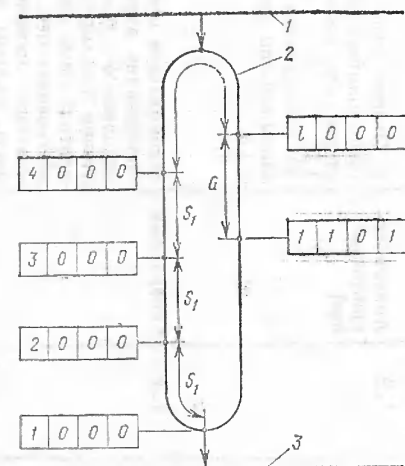


Рис. 10. Оптимальный вариант модифицированной равноинтервальной синхронизации адресов:

1 — канал ввода; 2 — накопительный регистр; 3 — канал вывода

Одностраничный способ имеет преимущества по сравнению с кратностраничным, а именно: меньшее время выборки за счет однократного обхода избыточных накопительных регистров, вводимых в структуру для повышения выхода годных микросхем; более простое преобразование адреса обращения; менее жесткие условия синхронизации адресов. Помимо изложенных выше рекомендаций, выбрать способ размещения информации надо с учетом минимальной информационной избыточности ЦМД-накопителя, обусловленной различием форматов НГМД и ЦМД-микросхем. В общем случае организация многокристального накопителя ЦМД-эмулятора может быть [10] последовательной (с одновременным доступом лишь к одной ЦМД-микросхеме накопителя); параллельной (с одновременным доступом ко всем ЦМД-микросхемам накопителя) и параллельно-последовательной (с одновременным доступом к нескольким ЦМД-микросхемам накопителя).

Характеристики способов организации накопителя позволяют оценить структурную избыточность накопителя ЦМД-эмулятора, обусловленную несоответствием форматов записей на диске и страниц ЦМД-микросхем, а также число ЦМД-микросхем, необходимых для эмуляции диска емкостью E .

Выбор оптимального способа организации накопителя целесообразно по критерию минимальной структурной избыточности из вариантов, удовлетворяющих требованиям системы по быстродействию, стоимости и др.

Выводы

1. Средства внешней памяти на ЦМД вступили в фазу промышленного освоения, открывая перспективу создания «бездисковых» ЭВМ, способных надежно работать в производственных цехах и в полевых условиях.

2. С учетом существующих и планируемых технико-экономических показателей ЦМД-ЗУ, целесообразные и емкие области их применения — системы числового программного управления станками и промышленными роботами, персональные ЭВМ.

3. На первом этапе серийного производства ЦМД-ЗУ при отсутствии программного обеспечения, учитывающего характерные режимы работы ЦМД-микросхем, один из наиболее быстрых и эффективных путей системной «привязки» ЦМД-ЗУ — эмуляция НГМД и в первую очередь НГМД. В последнем случае система воспринимает ЦМД-ЗУ как НГМД, обладающий, однако, существенно лучшими (в десятки раз) скоростными и надежностными характеристиками.

4. Для эмуляции НГМД применимы программный, аппаратный или

Сопоставление различных способов эмуляции

Классы ЦМД-эмуляторов	№ п/п	Способ преобразования адреса	Достоинства	Недостатки	Целесообразная область применения	
С внутренней эмуляцией	1.1	Программный	<p>а) Минимальные дополнительные затраты по сравнению с 1.2 и 1.3</p> <p>б) Возможность использования в накопителе микросхем с различной организацией</p> <p>в) Гибкость при перестройке на эмуляцию различных НГМД</p>	<p>При использовании контроллера с микропрограммой логика преобразования адреса не требует дополнительного оборудования и вводу, т.е. по сравнению с II отсутствует дублирование аппаратурных затрат</p>	<p>Увеличение времени доступа, снижающее общую производительность системы при организации программно - управляемого ввода-вывода и вводу</p>	<p>1. Память для мини-ЭВМ и микроЭВМ</p> <p>2. В устройствах с ПДЦ</p>
	1.2	Аппаратно-программный	<p>а) Относительно высокая производительность системы по сравнению с 1.1</p> <p>б) Относительно низкая стоимость по сравнению с 1.3</p>	<p>а) Необходимость замены ППЗУ при перестройке на эмуляцию другого НГМД</p> <p>б) Увеличение аппаратных затрат при массовом производстве по сравнению с 2.2 и 2.3</p>	<p>1. В устройствах, в которых показатель быстродействия доминирует над показателем стоимости</p> <p>2. Внешняя память для мини-ЭВМ</p>	
	1.3	Аппаратный	<p>Минимальное время преобразования адреса (определяемое быстродействием используемой элементной базы декодера), обеспечивающее максимальную производительность системы по сравнению с 1.1; 1.2; 2.1; 2.2 и 3.1</p>	<p>а) То же, что и в 1.2 а)</p> <p>б) Относительно высокая стоимость по сравнению с 1.2, 2.2 и 2.3</p>		
С внешней эмуляцией посредством специального контроллера ввода-вывода	2.1	Программный	<p>а) Минимальные дополнительные затраты (сравнимые с 1.1—1.3)</p> <p>б) Возможность работы с различными накопителями</p> <p>в) Гибкость при перестройке на эмуляцию другого НГМД</p>	<p>а) Возможность использования ЗУ на ЦМД в самостоятельной функции при соответствующем материальном обеспечении</p> <p>б) Гибкость при перестройке на эмуляцию другого НГМД, обеспечиваемая сме-</p>	<p>То же, что и в 1.1</p>	<p>1. В устройствах, в которых показатель быстродействия доминирует над показателем стоимости</p> <p>2. Внешняя память для микроЭВМ с перспективой перевода ЦМД-ЗУ на самостоятельное использование без адаптера</p>
				<p>Дублирование аппаратуры интерфейса в аппаратуре и в ЦМД-ЗУ</p>		

Классы ЦМД-эмуляторов	№ п/л.	Способ преобразования адреса	Достоинства	Недостатки	Целеобразная область применения
	2.2	Аппаратно-программный	<p>а) Относительно малое время преобразования по сравнению с 2.1</p> <p>б) Меньшие затраты при серийном производстве по сравнению с 1.2, 1.3 и 2.3</p>	<p>Наличие адаптера или его перепрограммирование</p> <p>Необходимость замены ПЗУ при изменениях в накопителе при перестройке на эмуляцию другого НГМД</p>	
	2.3	Аппаратный	<p>а) Максимальная производительность системы (по сравнению с 1.1, 1.2, 2.1, 2.2 и 2.3.1)</p> <p>б) Меньшие затраты при серийном производстве по сравнению с 1.2 и 1.3</p>	<p>Недостаточная гибкость при перестройке на эмуляцию другого НГМД</p>	
	3.1	Программный	<p>а) Минимизация по сравнению с остальными способами затраты на преобразование адреса</p> <p>б) Возможность использования ЗУ не только в режиме эмулятора</p> <p>в) Возможность работы с секторами произвольной длины</p>	<p>а) Снижение общей производительности системы (даже при использовании ПДП), обусловленное временными затратами на преобразование адреса</p>	<p>В системах памяти, в которых показатель стоимости доминирует над показателем быстродействия (персональные ЭВМ)</p>

программно-аппаратный методы эмуляции. Выигрыш в системах, для которых важен параметр стоимости, дают программный и программно-аппаратный методы. В системах, где важнее всего параметр быстродействия, наиболее приемлем аппаратный метод эмуляции.

5. Для перехода к «бездисковым» ЭВМ потребуется разработать специальные операционные системы, использующие преимущества ЦМД-ЗУ, увеличить форматированную емкость ЦМД-микросхем до 16...64 Мбит в корпусе. Это создаст условия для разработки ЦМД-ЗУ с емкостями 128...256 МБайт.

ЛИТЕРАТУРА

1. Kelly T. Designers differ over tapes, floppy discs and magnetic bubbles.—Canadian Controls + Instruments, 1982, v. 24, N 2, p. 49—52.

2. Маркаров Г. И., Раев В. К., Козлова Н. Д. ЗУ на ЦМД — уровень промышленного освоения.— В кн.: Радиоэлектроника в 1981 г. М.: Изд-во НИИЭИР, 1982, т. 2, с. 66—88.

3. Наумов Б. Н., Раев В. К., Маркаров Г. И.—Основные применения памяти на цилиндрических магнитных доменах для микроЭВМ.— Приборы и системы управления, 1981, № 10, с. 11—14.

4. Pierce R. Diskless computers emerge with proper mix of firmware, processor and bubble memory.—Electronic Design, 1983, v. 31, N 21, p. 177—182.

5. Козлова Н. Д., Раев В. К. Состояние и перспективы развития разработок ЗУ на ЦМД.—Радиоэлектроника (обзор НИИЭИР по материалам иностранной печати), 1984, 2, с. 14—77.

6. Bubble box holds megabyte.—Electronics, 1983, Feb., p. 236.

7. Тосц С. E. Le. Software driver lets CP/M address bubbles as a disk.—Electronics, 1981, N 3, pp. 143—148.

8. Кузнецов С. О., Падюков Л. Н.—В кн.: Научные труды ИНЭУМ. М.: ИНЭУМ, вып. № 104.

9. Wright W. E. Some file structure considerations pertaining to magnetic bubble memory.—The Computer J., 1983, v. 26, N 1, p. 43—51.

10. Cox G. Modular memories master X, Y, Z's of easy bubble-system expansion.—Electronic design, 1980, N 5, p. 103—106.

11. Ломов Л. С., Нурмухамедов Г. Н., Спивака Д. Д., Чиркин Г. К. Магнитная интегральная микросхема К1605РЦ1.—Электронная промышленность, 1983, № 4, с. 18—18.

13. Williams N. The adaptable bubble.—Systems International, 1984, May, p. 92—93.

13. Chester M.—Magnetic bubble memory update.—Computer Design, 1980, v. 19, N 5, p. 233—236, 238, 240.

Статья поступила 1 декабря 1984 г.

А. Н. Зимовец, К. С. Рыбак, С. В. Фомин, А. Ю. Черняк, А. Г. Юрочкин

МНОГОМАШИННАЯ СИСТЕМА ТЕСТИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Для настройки и проверки микропроцессорных средств вычислительной техники используются автономные рабочие места. В состав типового рабочего места (РМ) входит микроЭВМ, внешнее запоминающее устройство (ВЗУ) на гибких магнитных дисках (НГМД) для хранения тестового программного обеспечения, пультовой терминал и средства для подключения тестируемого оборудования.

Наличие ВЗУ на каждом РМ приводит к их удорожанию и низкой надежности из-за частых неисправностей НГМД и порчи магнитных носителей. Кроме того, ВЗУ в этом случае используются очень неэффективно: только для загрузки очередного теста. Для поддержания диалога с оператором, обслуживания файловой системы, запуска тестов и т. д. необходима операционная система, загружаемая из ВЗУ, компоненты которой занимают значительную часть оперативной памяти. В состав описываемой системы включена центральная ЭВМ (ЦМ) «Электроника МС 1212» (рис. 1), осуществляющая введение файловой системы и пересылку тестов в РМ по запросам, что обеспечивает оптимальное распределение функций между компонентами системы.

Этот подход позволяет упростить и повысить надежность РМ за счет исключения из их состава НГМД и уменьшения в несколько десятков раз объема программной поддержки на рабочих местах. Становится возмож-

ным в качестве единственного ВЗУ использовать высоконадежный, хотя и более дорогой накопитель. Этим сводятся к минимуму трудовые затраты на техническое обслуживание системы и поддержание качества магнитных носителей.

Структура системы. Система состоит из ЦМ и РМ, соединенных линиями связи (ЛС). Для достижения хорошей помехозащищенности, экономичности, нескритичности к пропускной способности обмен ведется последовательным старт-стопным кодом по интерфейсу «токовая петля 20 мА». Тесты хранятся в ВЗУ и передаются на рабочие места в абсолютном формате, что обеспечивает возможность надежного контроля получаемой информации, простоту алгоритма декодирования и загрузки тестов в оперативную память микроЭВМ. Кроме того, обеспечивается простота расширения библиотеки тестов, так как трансляторы с большинства языков программирования и редакторы связей практически всех операционных систем позволяют получать рабочие программы в абсолютном формате.

Для упрощения системы выбран сравнительно несложный протокол обмена между РМ и ЦМ. Сеанс связи начинается по инициативе оператора передачей в ЦМ однобайтового запроса на пересылку конкретного теста. ЦМ отыскивает в библиотеке необходимый тест и пересылает его в микроЭВМ рабочего места. На этом сеанс связи заканчивается. Если указанного теста в библиотеке нет,

то в микроЭВМ передается специальная байтовая посылка, которая не может быть первой при пересылке теста, и сеанс связи немедленно прекращается.

Рабочее место. В состав РМ входит процессор, программно-совместимый с микроЭВМ «Электроника 60», блок ОЗУ, средства для подключения тестируемого оборудования, двухканальное устройство последовательного обмена (И12) для подключения пультового терминала и связи с ЦМ и ПЗУ для хранения программы обслуживания РМ.

Программа, написанная на языке ассемблера, обеспечивает высокую скорость размещения в оперативной памяти принятой из ЛС информации, что необходимо из-за отсутствия обратной связи с ЦМ при передаче теста, занимает всего 126 слов памяти, размещается в ПЗУ в области адресов регистров внешних устройств.

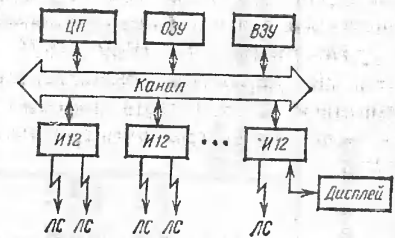


Рис. 2. Структура связей центральной микроЭВМ системы

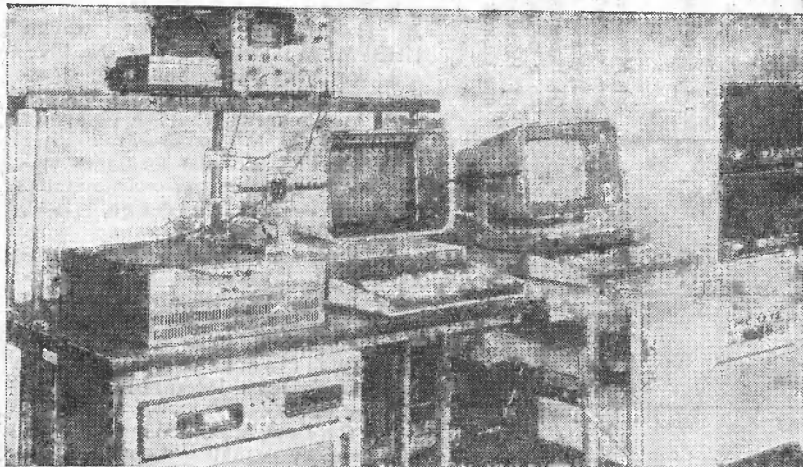


Рис. 1. Система тестирования микропроцессорных средств на основе микроЭВМ «Электроника МС 1212»

После запуска программа запрашивает мнемонику теста у оператора, передает ее в ЛС и переходит к приему информации от ЦМ. При получении сигнала об отсутствии теста возобновляется диалог с оператором, в противном случае производится прием и размещение теста в оперативной памяти. Программа обеспечивает контроль вводимой оператором мнемоники и совпадение контрольных сумм блоков принимаемого из ЛС теста.

Центральная ЭВМ. В состав ЦМ входит процессор, оперативная память, накопитель на сменных магнитных дисках СМ-5400 и двухканальные интерфейсы для подключения пультового терминала и соединения с необходимым числом РМ (рис. 2).

Президиум Верховного Совета СССР назначил тов. Колесникова Владислава Григорьевича министром электронной промышленности СССР.

КОЛЕСНИКОВ ВЛАДИСЛАВ ГРИГОРЬЕВИЧ

Тов. КОЛЕСНИКОВ В. Г., 1925 года рождения, русский, член КПСС с 1961 года. Окончил Воронежский политехнический институт. Кандидат технических наук, член-корреспондент Академии наук СССР.

Трудовой путь начал в 1941 году слесарем-сборщиком на оружейном заводе; затем работал техником, инженером, заместителем главного конструктора завода радиодеталей. С 1958 года — на заводе полупроводниковых приборов: главным технологом, начальником конструкторского бюро, главным инженером и директором. В 1969—1971 годах — генеральный директор производственного объединения. С 1971 года работал первым заместителем министра электронной промышленности.



Герой Социалистического Труда; награжден двумя орденами Ленина, орденами Октябрьской Революции и «Знак Почета», медалями. Лауреат Ленинской премии и Государственной премии СССР.

«Известия», 1985 г., 19 ноября

В качестве операционной среды ЦМ выбрана ОС Рафос [1], отличающаяся простотой, компактностью и быстротой реакции на внешние воздействия. В то же время эта ОС предоставляет достаточные возможности для реализации системы и поддержки ее дальнейшего развития.

Программа ЦМ написана на языке Паскаль [2]. В нее включены средства обслуживания ЛС на физическом уровне. Это позволяет снять ограничения многотерминального драйвера ОС на число подключаемых ЛС, ускорить обмен и упростить настройку системы на заданное число РМ и применение нестандартных интерфейсов связи. Работа с интерфейсами ЛС осуществляется по опросу «флага». Работа по прерыванию выгодна в тех случаях, когда ЭВМ должна переключаться с основного процесса обработки информации на обработку запроса. В данной задаче обслуживание ЛС и является основ-

ным процессом, поэтому использование обмена с ЛС по опросу флага позволяет значительно упростить структуру программы и исключить неизбежные при работе по прерыванию потери времени на переключение контекстов.

Для уменьшения объема аппаратного обеспечения ЦМ целесообразно вместо автономных двухканальных интерфейсов использовать простые мультиплексоры передачи данных.

Заключение. При скорости обмена в ЛС 9600 бод время передачи одного теста в системе, включающей 15 рабочих мест, составляет от 5 до 15 с в зависимости от его объема, а среднее время ожидания оператором РМ загрузки теста — 20 с.

Структура программного обеспечения ЦМ позволяет наращивать возможности системы, не затрагивая широко тиражируемое аппаратное и программное обеспечение РМ. Преду-

сматривается создание версии программы для ЦМ с библиотечкой тестов, размещаемой в расширенной оперативной памяти и загружаемой с диска однократно при запуске системы. Это исключит обращения к ВЗУ в процессе работы, повысит скорость передачи тестов и надежность системы. Для уменьшения времени реакции предусмотрена параллельная пересылка тестов в несколько РМ.

ЛИТЕРАТУРА

1. Операционная система СМ ЭВМ РАФОС: Справочник / Под общ. ред. В. П. Семика. — М.: Финансы и статистика, 1984.
2. Йенсен К., Вирт Н. Паскаль: Руководство для пользователя и описание языка: Пер. с англ. — М.: Финансы и статистика, 1982.

Статья поступила 28 мая 1985 г.

Увеличить производство программных средств для вычислительной техники и автоматизированных систем управления.

Из проекта Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года

УДК 681.3.06

С. С. Лавров, А. О. Слисенко, Г. С. Цейтин

ПРОЕКТ ПЛАНА-ПРОГРАММЫ ПО СПЕЦИАЛЬНОСТИ «ИНФОРМАТИКА И СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ»

Вероятно, читателей журнала не нужно убеждать в необходимости подготовки в вузах специалистов по программному обеспечению ЭВМ. Однако, как ни парадоксально, вузы таких специалистов до сих пор не выпускают: в номенклатуре вузов такой специальности попросту нет. Иногда, впрочем, она фигурирует в качестве разновидности какой-нибудь другой специальности, например прикладной математики (код 0647).

В нынешней ситуации, когда резко возрастает применение вычислительной техники во всех сферах нашей жизни и промышленные предприятия испытывают особенно острую нужду в квалифицированных специалистах этого профиля, отсутствует даже возможность обратиться к высшей школе с заказом на такого специалиста. А предложения промышленности об увеличении выпуска специалистов по прикладной математике приводят к тому, что кафедры, отвечающие за эту специальность, специализируют студентов в соответствии со сложившейся структурой прикладной математики, и в результате увеличивают выпуск совсем не тех специалистов.

Кроме того, в рамках «чужой» специальности нет возможности обеспечить преподавание необходимых для программирования дисциплин в должном объеме, так как учебный план специализации может отклоняться от «основного» плана не более, чем на 10 %.

Такое положение программирования в определенной степени связано с бытующим взглядом на программиста как на малоквалифицированного ремесленника, который кодирует для машины, что ему зададут. Некоторые считают, что с распространением всеобщей компьютерной грамотности специальность программиста вовсе исчезнет. При этом забывают, что работа с ЭВМ может стать доступной для непрофессионалов лишь в том случае, если предварительно усилиями высококвалифицированных программистов будет создано необходимое программное обеспечение. Как «первая грамотность», упразднив профессию писаря, не отменила профессии литератора и даже повысила ее роль, так и «вторая грамотность» не отменит профессию разработчика программных систем, а напротив, повысит требования к ней.

Распространено также мнение, что программирование как научная дисциплина не существует и не имеет права на существование, что оно растворяется в таких проблемах, как создание математических моделей разнообразных явлений и процессов и разработка общей структуры и архитектуры ЭВМ. Однако на самом деле программы решения самых разных задач имеют много общего по своей внутренней структуре, организации, характеру функционирования. Это значит, что программирование порождает ряд понятий, методов и приемов

работы, связанных именно с программами как объектом деятельности, а не только как средством решения исходных задач. Программирование породило достаточно обширную и сложную проблематику, тесно связанную со многими разделами математики. Но программа — это все-таки не абстрактный, а вполне реальный объект, поэтому программирование не стало математической дисциплиной, а отпочковалось от математики, подобно тому, как это ранее произошло с механикой, теоретической физикой и т. п. Современное программирование — это достаточно развитая и сложная область научной деятельности, в которой получен ряд фундаментальных результатов, имеющих принципиальное значение для практики.

Авторы настоящей статьи, предлагая введение программирования в качестве самостоятельной вузовской специальности и составляя публикуемый здесь проект учебного плана, исходили из того, что подготовка студентов по этой специальности должна опираться на собственный фундамент, следовать собственной логике изложения. Этому требованию был подчинен отбор включенных в план общематематических и прикладных дисциплин и их место в общей программе. Некоторые коллеги сочли предлагаемый план недостаточным, так как остались неосвещенными отдельные важные направления и методы. К сожалению, узкие рамки пятилетнего обучения не позволяют расширить объем сообщаемых сведений, да в этом и нет необходимости. Можно варьировать предлагаемый план, изменять набор спецкурсов и спецсеминаров, сохраняя тот фундамент необходимых программисту знаний и навыков, который предлагается заложить на протяжении первых 5—6 семестров.

Следует отметить, что в эти основные курсы иногда включался и материал, который традиционно относят к тем или иным специальным дисциплинам. Такой выбор диктовался не стремлением дать предварительное знакомство с этими дисциплинами, а логикой изложения самих начальных курсов. Например, изучение методов трансляции на втором курсе имеет целью не подготовку к разработке трансляторов, а освещение взаимоотношений между программой на языке высокого уровня и программой в кодах машины. Аналогично, примеры на моделирование не служат введением в курс имитационного моделирования, а направлены на демонстрацию языковых средств программирования на материале ситуаций моделирования и роли моделирования как одного из важных приемов программирования.

В предлагаемом плане не отражены учебные курсы общеобразовательного характера, такие, как иностранные языки, физическая подготовка и общественно-политические дисциплины. Предполагается, что программы

по этим курсам и их распределение по семестрам остаются такими же, как и у большинства других специальностей. Впрочем, в отношении гуманитарных дисциплин это решение не бесспорно. Информатика, рассматриваемая как самостоятельная область знаний, не может обойтись без учета таких факторов, как связь программного обеспечения со структурой человеческого знания, с мышлением, языком, как общественный характер знания и общественная роль информации (в том числе программного обеспечения), как неизбежно коллективный характер программных разработок. Так, что, вероятно, специалисты интересующего нас профиля нуждаются и в углубленном изучении некоторых вопросов диалектического материализма, языкознания, психологии, социологии и т. п. Однако пока не разработаны соответствующие учебные программы.

Настоящая программа по специальности «Информатика и системное программирование» содержит ядро — курсы, составляющие обязательный минимум по алгоритмике и программированию. Они отмечены словами «обяз. мин.», стоящими справа от названия этих курсов. В остальной части программа ориентирована на университет и имеет целью дать и хорошую общематематическую подготовку.

В варианте для технических вузов можно сократить общематематические курсы первых семестров, но обязательным становится курс физики с акцентом на электродинамику и физику твердого тела и с практическими и лабораторными занятиями. В вузах с 11-семестровым обучением можно также углубить изучение аппаратуры вычислительных машин и конкретных программных систем.

Составители исходили из реального числа учебных недель. Условно принято, что в семестре 16 учебных недель. Это означает, что, например, 48 ч занятий подразумевает среднее число часов в неделю, равнос 3.

В тексте приняты следующие сокращения:

- лк — лекционный курс,
- пз — практические занятия,
- дс — работа за дисплеем с преподавателем или самостоятельная,
- лб — лабораторные занятия.

В рамках данной специальности предлагаются следующие специализации:

- а) разработка и эксплуатация системного программного обеспечения;
- б) разработка и эксплуатация программного обеспечения для управления техническими средствами;
- в) разработка и эксплуатация программного обеспечения для автоматизации научных исследований и конструкторских разработок;
- г) разработка и эксплуатация программного обеспечения для автоматизации планирования и управления.

Возможные специализации различаются спецкурсами и спецсеминарами, примерный перечень которых приведен в конце данной программы. После названия спецкурса стоят буквы а, б, в или г, указывающие, для каких из перечисленных специализаций предназначен данный курс.

Для высокоэффективного использования вычислительных машин необходима алгоритмическая, языковая и технологическая культура. Причем алгоритмическая культура носит наиболее фундаментальный характер — ведь в вычислительной машине работает алгоритм. Развитие этой культуры у студентов — наиболее трудная задача обучения. Необходимость языковой и технологической культуры определяется тем, что программное обеспечение разрабатывается людьми и для людей. Только комплексное, всестороннее теоретическое и практическое изучение предлагаемых дисциплин, объединяющее все три упомянутые компоненты вычислительной культуры, при условии их хорошей сбалансированности, может дать полноценного специалиста по применению вычислительной техники.

Без практической работы с современной периферией

(дисплеи, графопостроители, таблетки и т. д.) невозможна подготовка специалистов высокого уровня. В программе явно указана (если не считать курса машинной графики) только работа за дисплеем как наиболее важная и ресурсоемкая часть освоения периферии и собственно программирования. По данной программе среднее дисплейное время на одного студента составляет около 9 ч в неделю. Одна учебная группа из 25 человек требует в среднем $25 \times 9 = 225$ ч в неделю. Далее надо иметь в виду, что значительное дисплейное время требуется преподавателям, по-видимому, тоже около 8—10 ч в неделю на человека, а также нужно время для факультативных занятий студентов. По понятным причинам необходимо иметь по крайней мере 2 дисплейных класса, причем на разной технике. Вероятно, разумно ориентироваться на небольшие, по 10—11 дисплеев классы, и для дисплейных занятий с преподавателем разбивать учебные группы на три части.

В программе умышленно не упомянуты конкретные языки программирования. Это сделано потому, что они постоянно развиваются, и, что более существенно, в одной и той же ситуации можно применять различные языки. Например, в практической части курса «Основы программирования» можно использовать, скажем, Паскаль, Алгол-68 или ПЛ/1 (введя соответствующий материал в лекционный курс). Предполагается, что к концу обучения студенты осваивают около 10 распространенных языков программирования.

1-й СЕМЕСТР (1-й курс)

ОСНОВЫ ДИСКРЕТНОЙ МАТЕМАТИКИ (обяз. мин.)

32лк + 32пз

зачет, экзамен

А. Лекционный курс

Слово. Алфавит. Список. Операции над словами и списками. Примеры.

Множества и функции. Операции над множествами и функциями. Полные и частичные функции. Примеры. Конечные множества. Способы задания конечных множеств.

Булевские функции. Представление булевских функций формулами. Нормальные формы.

Конечные графы. Способы задания. Степень. Пути. Связность. Деревья. Корень. Предки. Потомки. Упорядоченные деревья. Циклы. Остовы.

Примеры. Задачи, формулируемые в терминах циклов, путей, остовов. Представление булевских функций комбинационными схемами.

Логико-предметные формулы. Синтаксис. Структурное дерево. Префиксная, инфиксная и постфиксная формы. Представление множеств и графов формулами. Другие примеры. Гиперграфы.

Алгебраическая система. Выполнимость формул на алгебраической системе.

Конечные автоматы. Различные формы задания и примеры.

Машина Тьюринга. Задание подстановками и системой команд.

Б. Практические занятия

Представление данных различных задач в виде слов. Двоичное и алфавитное представление натуральных чисел.

Способы задания булевских функций. Приведение к нормальным формам. Эквивалентные преобразования. Представление схемами.

Техника работы с графами. Построение кратчайших путей, эйлеровых и гамильтоновых циклов.

Построение формул, соответствующих основным конструкциям естественного языка. Простейшие эквивалентные преобразования.

Построение простых машин Тьюринга и конечных автоматов.

ОСНОВЫ ПРОГРАММИРОВАНИЯ (обяз. мин.)

32лк + 32пз + 32дс

зачет с оценкой

А. Лекционный курс

Общая схема решения задачи на вычислительной машине. Демонстрация на примерах основных этапов решения задач. Краткое описание основных функциональных блоков машины. Работа оператора.

Общая структура программного обеспечения. Операционная система с точки зрения пользователя. Язык управления заданиями. Интерактивный режим. Пользовательский редактор текстов.

Программирование на языке высокого уровня. Основные конструкции и структуры данных, обеспечивающие структурность программирования.

Обращение к файлам в языке программирования.

Приемы моделирования реальных процессов на машине.

Б. Практические занятия

Программирование простейших задач, как то: действия над векторами и матрицами, простейшая статистическая обработка данных, задачи на графах, обработка строк, моделирование автоматов. Объяснение работы составленных программ. Структурность и понятность программы. Количественные характеристики программы и качество ее функционирования. Зависимость этих характеристик от исходного алгоритма и способа реализации.

В. Работа за дисплеем

Устройство дисплея. Основы машинописи. Культура работы за дисплеем. Освоение текстового редактора.

Подготовка программы к работе на вычислительной машине.

Отладка программ и запуск их на выполнение. Анализ результатов.

МАТЕМАТИЧЕСКИЙ АНАЛИЗ

64лк + 48пз

зачет, экзамен

А. Лекционный курс

Числа, функции, отображения. Индукция. Аксиоматическое определение вещественных чисел. Счетные множества. Многомерное вещественное пространство. Неравенство Коши — Буяковского.

Предел последовательности. Основные свойства предела.

Дифференциальное исчисление в многомерном вещественном пространстве. Непрерывность. Основные свойства непрерывных отображений. Глобальные свойства непрерывных функций. Классические теоремы.

Дифференциал отображения. Единственность. Дифференциал суммы, произведения, суперпозиции, обратной функции (для случая одной переменной). Геометрический смысл дифференциала.

Понятие первообразной. Интеграл непрерывной функции одной переменной. Основные свойства. Единственность интеграла.

Локальные свойства функций одной переменной. Многочлены Тейлора. Формула Тейлора с различными остаточными членами.

Функции многих переменных. Вычисление дифференциала. Матрица Якоби. Градиент. Условия экстремума. Формула Тейлора для функций многих переменных.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на алгоритмику и вычислительные приложения (в частности, подготовка к работе с системами символьных вычислений).

АЛГЕБРА И ГЕОМЕТРИЯ

64лк + 48пз

зачет, экзамен

А. Лекционный курс

Теория делимости целых чисел. Алгоритм Евклида. Простые числа. Теория сравнений. Теорема Эйлера. Китайская теорема об остатках.

Некоторые общие понятия алгебры. Группы, подгруппы, кольца, поля. Изоморфизм.

Комплексные числа. Тригонометрическая форма. Корни из единицы. Первообразные корни. Показательная и логарифмическая функции.

Элементы аналитической геометрии. Векторы на плоскости и в пространстве и операции над ними. Декартовы координаты. Аффинное преобразование координат. Уравнения прямой и плоскости. Расстояние и углы между прямыми и плоскостями. Выпуклые множества на плоскости и в пространстве.

Элементы алгебры полиномов. Полиномы от одной переменной. Схема Горнера и теорема Безу. Число корней. Алгебраически замкнутое поле. Полиномы от многих переменных.

Матрицы и определители. Векторные пространства. Операции над матрицами. Определитель и его свойства. Вычисление определителей. Теорема Крамера.

Линейная зависимость. Базис и ранг. Системы линейных уравнений общего вида. Обращение квадратных матриц. Ортогональные матрицы. Характеристический полином.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на алгоритмические вопросы. Рекурсивные алгоритмы для умножения длинных чисел. Модульная арифметика. Умножение 2×2 -матриц с 7 умножениями. Построение выпуклой оболочки плоского множества. Переход от аффинных координат к полярным и обратно.

(Общее время перечисленных занятий в 1-м семестре: всего 384 ч, в неделю 24 ч)

(Дисплейное время в 1-м семестре: всего 32 ч; в неделю 2 ч на одного студента)

2-й СЕМЕСТР (1-й курс)

ТЕОРИЯ АЛГОРИТМОВ (обяз. мин.)

32лк + 32пз

зачет, экзамен

А. Лекционный курс

Алгоритмы как математическая модель вычислительного процесса. Машина с адресным доступом к памяти. Программа. Процесс. Сложностные характеристики вычислений. Примеры. Свойства замкнутости алгоритмов. Интерпретатор.

Основные теоремы теории алгоритмов и их программистский смысл. Универсальный алгоритм. Алгоритмическая неразрешимость проблемы самоприменимости. Теорема Клини о нормальной форме. Итерационная теорема. Алгоритмическая неразрешимость проблемы равенства полных алгоритмов. Теорема о рекурсии. Смешанные вычисления. Применения теоремы о рекурсии.

Перечислимые и разрешимые множества.

Алгоритмическая неразрешимость общих задач оптимизации программ и общих задач символьных вычислений.

Б. Практические занятия

Построение простых алгоритмов и их сложностной анализ. Доказательство алгоритмической неразрешимости простейших задач по схеме, изложенной в лекционном курсе. Применения теоремы о рекурсии к построению алгоритмов.

ОСНОВЫ ПРОГРАММИРОВАНИЯ (обяз. мин.)

32лк + 16пз + 32дс

зачет с оценкой

А. Лекционный курс

Членение программы на модули.

Процедуры. Интерфейс процедуры с вызвавшей ее программой. Передача параметров. Рекурсивные вызовы.

Поддержка модульности операционной системой.

Реакция на исключительные ситуации.

Спецификации на логическом языке. Инварианты цикла. Проблема соответствия программы спецификациям.

Программирование «снизу вверх» и «сверху вниз». Методы отладки программ.

Б. Практические занятия

Программирование на языке высокого уровня. Параллельно осваивается конкретная алгоритмика по курсу «Основы дискретной математики».

В. Работа за дисплеем

Отладка программ, составленных на практических занятиях. Самостоятельное программирование.

МАТЕМАТИЧЕСКИЙ АНАЛИЗ

64лк + 32пз зачет, экзамен

А. Лекционный курс

Ряды. Признаки сходимости. Степенные ряды. Радиус сходимости. Дифференцируемость. Элементарные функции.

Функции комплексного переменного. Производная. Конформные отображения. Интеграл. Интегральная формула Коши. Разложение в степенной ряд. Нули. Ряд Лорана. Целые и мероморфные функции. Вычеты. Аналитическое продолжение.

Несобственные интегралы и интегралы, зависящие от параметра. Перестановочность с предельным переходом. Дифференцирование по параметру. Асимптотические разложения.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на алгоритмику и материал, необходимый для работы с системами символьных вычислений.

АЛГЕБРА И ГЕОМЕТРИЯ

64лк + 48пз зачет, экзамен

А. Лекционный курс

Кривые и поверхности 2-го порядка. Квадратичные формы. Преобразование квадратичной формы к каноническому виду линейной подстановкой переменных. Закон инерции. Ортогональные преобразования квадратичной формы к каноническому виду. Эрмитовы формы.

Полиномы и дроби. Теория делимости полиномов от одной переменной. Рациональные дроби. Интерполяция. Сравнения в кольце полиномов и расширения полей. Полиномы от одной переменной над факториальными кольцами.

Основная теорема о существовании корней. Распределение корней. Обобщенная теорема Штурма.

Элементы теории групп. Группы преобразований. Конечные абелевы группы.

Элементы теории чисел. Символы Якоби и Лежандра. Закон взаимности. Индекс. Проблема проверки простоты и разложения на множители. Простейшие арифметические и алгебраические коды.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на алгоритмику.

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

32дс входит в зачет по курсу «Основы программирования»

Работа за дисплеем

Работа с системами символьных вычислений. Решение задач по курсам «Математический анализ» и «Алгебра и геометрия».

(Общее время перечисленных занятий во 2-м семестре: всего 384 ч, в неделю 24 ч)

(Дисплейное время во 2-м семестре: всего 64 ч, в неделю 4 ч на одного студента)

3-й СЕМЕСТР (2-й курс) СТРУКТУРЫ ДАННЫХ (обяз. мин.)

32лк + 16пз + 32дс зачет, экзамен

А. Лекционный курс

Машинное представление структур данных языков высокого уровня. Работа с адресами.

Массивы. Структуры (записи). Строки. Списки. Стеки. Очереди. Словари. Деревья.

Структуры данных во внешней памяти.

Процедуры и процессы как данные. Абстрактные типы данных.

Б. Практические занятия

Основные алгоритмы работы со структурами данных. Использование структур данных в конкретных программах.

В. Работа за дисплеем

Отладка программ, составленных на практических занятиях. Самостоятельное программирование.

МАШИННЫЕ ЯЗЫКИ (обяз. мин.)

32лк + 32пз + 16дс зачет, экзамен

А. Лекционный курс

Представление данных в вычислительной машине. Типы команд. Регистры. Способы адресации данных. Арифметические и логические операции. Обращение к памяти. Разветвления.

Защита системы. Привилегированные команды. Прерывания. Команды управления периферийными устройствами.

Аппаратурная поддержка языковых средств высокого уровня. Стеки. Очереди. Процедуры. Виртуальная память. Теги. Дескрипторы.

Язык ассемблера.

Б. Практические занятия

Составление небольших программ, углубляющее и конкретизирующее материал лекционного курса.

В. Работа за дисплеем

Отладка небольших программ на ассемблере.

МАТЕМАТИЧЕСКИЙ АНАЛИЗ

64лк + 32пз зачет, экзамен

А. Лекционный курс

Краткое изложение теории интеграла. Измеримые множества. Суммы Дарбу. Теорема Фубини.

Интегрирование дифференциальных форм. Теоремы о неявном и обратном отображении.

Поверхности. Явное задание поверхностей. Ориентация. Теорема Стокса.

Элементы дифференциальной геометрии. Векторные функции скалярного аргумента. Основной трехгранник. Кривизна и кручение. Внутренние координаты поверхности. Измерение длин, углов и площадей. Главные кривизны, полная и средняя кривизна поверхности.

Б. Практические занятия

Занятия по лекционному курсу. Основное внимание уделяется геометрическим и вычислительным приложениям.

АЛГЕБРА

64лк + 32пз зачет, экзамен

А. Лекционный курс

Векторные пространства. Линейные и полилинейные отображения. Линейные операторы (комплексный и вещественный случаи). Элементы алгебры тензоров.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на алгоритмические вопросы, в частности, касающиеся применений систем символьных вычислений.

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

32дс входит в зачет по курсу «Основы программирования»

Работа за дисплеем

Работа с системами символьных вычислений. Решение задач по курсам «Математический анализ» и «Алгебра».

(Общее время перечисленных занятий в 3-м семестре: всего 384 ч, в неделю 24 ч)
(Дисплейное время в 3-м семестре: всего 80 ч, в неделю 5 ч на одного студента)

4-й СЕМЕСТР (2-й курс)

ОСНОВЫ ТРАНСЛЯЦИИ (обяз. мин.)

64лк + 32пз + 32дс

зачет, экзамен

А. Лекционный курс

Интерпретация и компиляция. Отображение конструкций высокого уровня в машинные команды. Использование регистров и косвенной адресации. Распределение памяти. Поддерживающие программы периода выполнения. Связь с операционной системой.

Применение грамматик к описанию синтаксиса языков программирования.

Бесконтекстные языки и грамматики. Линейные языки. Языки с ограниченным числом проходов.

Автоматные языки. Автоматы для анализа грамматик: магазинные и конечные. Автоматные преобразователи.

Проектирование компиляторов. Число проходов. Проблема межмашинных языков. Объектные языки.

Таблицы символов и видов.

Лексический анализ.

Оптимизация. Распределение регистров.

Исправление и диагностика ошибок.

Проблема надежности.

Б. Практические занятия

Применение грамматик для описания фрагментов языков. Освоение техники анализа грамматик. Проектирование учебного компилятора. Выработка навыков коллективной работы.

В. Работа за дисплеем

Реализация учебного компилятора по фрагментам.

КОМБИНАТОРНЫЕ АЛГОРИТМЫ (обяз. мин.)

48лк + 16пз

экзамен

А. Лекционный курс

Комбинаторная техника анализа алгоритмов. Конечные разностные уравнения и неравенства. Приближенные формулы для факториала и биномиальных коэффициентов. Перестановки.

Основные алгоритмы сортировки и их сложностной анализ.

Сбалансированные деревья и их использование в алгоритмах информационного поиска.

Хеширование. Методы построения хеш-функций. Настройка хеш-функций.

Представление вхождений деревьями идентификаторов. Поиск подслов из регулярного множества.

Алгоритмы поиска сначала-вглубь и сначала-вширь. Нахождение связанных и сильно связанных компонент графа. Построение транзитивного замыкания графа. Нахождение доминаторов.

Б. Практические занятия

Построение и практический анализ алгоритмов на базе конструкций лекционного курса. Основное внимание уделяется практическим аспектам применений сбалансированных деревьев, хеширования и поиска по образцу.

МАТЕМАТИЧЕСКИЙ АНАЛИЗ

32лк + 16пз

зачет, экзамен

А. Лекционный курс

Элементы гармонического анализа. Полные и замкнутые системы. Ряды Фурье. Сходимость. Приложения.

Интеграл Фурье. Задача о приближенном суммировании рядов Фурье. Преобразования Фурье и Лапласа. Приложения.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на вычислительные аспекты, связанные с приложениями.

ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ

32лк

зачет с оценкой

Лекционный курс

Основные понятия. Общие теоремы о существовании и единственности решений. Простейшие понятия об устойчивости решений.

Обыкновенные дифференциальные уравнения. Постановка задач, связанные с решением обыкновенных дифференциальных уравнений. Решение простейших уравнений. Простейшие методы приближенного решения. Теоремы существования и единственности решения задачи Коши для уравнений первого порядка.

Системы линейных обыкновенных дифференциальных уравнений 1-го порядка.

Понятие о первых интегралах.

Элементы качественной теории.

Постановки краевых задач.

ТЕОРИЯ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКАЯ СТАТИСТИКА

48лк + 16пз

зачет, экзамен

А. Лекционный курс

Конечные поля вероятностей. Урновые схемы. Понятие о проверке статистических гипотез.

Элементы комбинаторного анализа. Основные комбинаторные функции. Приложения. Гипергеометрическое распределение. Примеры. Биномиальное распределение и распределение Пуассона.

Случайные величины. Независимость. Математические ожидания. Производящие и характеристические функции. Дисперсии. Ковариации.

Неравенство Чебышева и его уточнения. Центральная предельная теорема. Теорема Лапласа. Законы больших чисел.

Применения центральной предельной теоремы к задаче различения гипотез и построению интервальных оценок.

Теорема Пуассона. Оценка для отклонения биномиальных вероятностей от пуассоновских.

Непрерывные распределения. Нормальное распределение.

Анализ простейших случайных блужданий и ветвящихся процессов.

Простейшие случайные процессы. Пуассоновский процесс.

Марковские процессы.

Составление уравнений для переходных вероятностей в процессах массового обслуживания.

Элементы теории информации. Источники, кодирование, энтропия и количество информации.

Б. Практические занятия

Занятия по лекционному курсу с акцентом на приложения к статистическому анализу информационных и вычислительных систем.

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

48дс

зачет

Работа за дисплеем

Решение задач по курсам «Комбинаторные алгоритмы», «Дифференциальные уравнения», «Теория вероятностей и математическая статистика».

(Общее время перечисленных занятий в 4-м семестре: всего 384 ч, в неделю 24 ч)

(Дисплейное время в 4-м семестре: всего 80 ч, в неделю 5 ч на одного студента)

АНАЛИЗ ВЫЧИСЛЕНИЙ (обяз. мин.)

32лк + 32пз зачет, экзамен

А. Лекционный курс

Методика построения математических моделей вычислений.

Распределенные вычисления. Интерфейс и протокол с точки зрения теории алгоритмов. Примеры.

Сравнение вычислительной силы различных моделей. Сложностная классификация задач. Сложностная иерархия.

Доказуемо трудные задачи. Задачи с экспоненциальными и более высокими нижними оценками. Условно трудные задачи. Полные переборные задачи.

Обзор общих методов сокращения перебора.

Б. Практические занятия

Сложностной анализ алгоритмов, возникающих при постановке прикладных задач.

ЯЗЫКИ И СИСТЕМЫ ПРОГРАММИРОВАНИЯ

(обяз. мин.)

32лк экзамен

Лекционный курс

Современный взгляд на методологию программирования.

Общая организация языков программирования. Уровневая структура систем программирования. Типы языков, назначение языков (машинно-ориентированные и проблемно-ориентированные языки, языки системного программиста, прикладного программиста и конечного пользователя).

Вопросы синтаксического оформления языков программирования.

Структуры доступа (блочная организация программы, область действия идентификаторов, подключение пакетов процедур, динамическая идентификация).

Структуры управления. Параллельные процессы.

Программное окружение.

Сопряжение с операционной системой. Реакции на исключительные ситуации.

Понятие о пакетах прикладных программ.

ОПЕРАЦИОННЫЕ СИСТЕМЫ (обяз. мин.)

48лк + 32дс зачет с оценкой

А. Лекционный курс

Функции и основные принципы построения операционных систем. Процессы, их организация и синхронизация. Распределение времени процессора.

Управление памятью. Виртуальная память.

Управление ресурсами. Системные тупики.

Защита операционной системы. Учетные системы.

Интерактивный режим. Опознание и взаимная защита пользователей.

Проектирование операционных систем. Настройка операционной системы на конкретную вычислительную машину. Оценка эффективности.

Примеры операционных систем.

Б. Работа за дисплеем

Освоение современных операционных систем. Анализ их работы. Учебные модификации операционных систем.

ФУНКЦИОНАЛЬНЫЙ АНАЛИЗ

32лк зачет с оценкой

Лекционный курс

Нормированные и гильбертовы пространства. Полнота. Теоремы о неподвижной точке. Приложения к приближенному решению уравнений.

Функционалы. Вычисление нормы интегральных операторов.

Ортогональные ряды. Приложения к построению и анализу приближенных алгоритмов.

32лк

зачет с оценкой

Лекционный курс

Классификация уравнений в частных производных. Задача Коши. Теоремы о существовании и единственности решения.

Понятие о характеристике.

Приведение к каноническому виду и классификации уравнений 2-го порядка.

Гиперболические уравнения.

Эллиптические уравнения.

Параболические уравнения.

Вариационные методы решения задач математической физики.

ОСНОВЫ ПРИБЛИЖЕННЫХ ВЫЧИСЛЕНИЙ

48лк + 16дс

экзамен

А. Лекционный курс

Общие понятия. Ошибки округления. Сложность вычислений и устойчивость алгоритмов.

Вычисление арифметических выражений. Вычисление значений полиномов. Быстрое преобразование Фурье.

Решение линейных систем над различными полями и кольцами.

Поиск корней многочленов.

Приближение функций многих переменных. Интерполяция.

Составные кривые и сплайны.

Составные поверхности.

Численное интегрирование.

Б. Практические занятия

Решение задач по лекционному курсу. Освоение соответствующих программных систем.

ТЕОРИЯ ВЕРОЯТНОСТЕЙ
И МАТЕМАТИЧЕСКАЯ СТАТИСТИКА

48лк

экзамен

Лекционный курс

Методы статистического описания. Статистические решения.

Последовательный анализ. Точечные и интервальные оценки.

Регрессионный анализ и метод наименьших квадратов.

Общая теория проверки гипотез. Непараметрические методы, порядковые статистики. Статистические задачи метода случайных испытаний. Моделирование случайных величин с заданными распределениями. Вероятностные алгоритмы. Вероятностные оценки функционирования алгоритмов (в том числе, оценки сложностных характеристик).

Стационарные процессы. Понятие спектра. Фильтры. Оценка спектра по результатам наблюдений.

ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ

32дс

зачет

Работа за дисплеем

Решение задач по курсам «Основы приближенных вычислений» и «Теория вероятностей и математическая статистика».

КУРСОВАЯ РАБОТА (начало)

48дс

СПЕЦКУРСЫ (факультативно)

(Общее время перечисленных занятий в 5-м семестре, не считая времени на курсовую работу и спецкурсы: всего 384 ч, в неделю 24 ч)

(Дисплейное время в 5-м семестре: всего 128 ч, в неделю 8 ч на одного студента)

МАТЕМАТИЧЕСКАЯ ЛОГИКА (обяз. мин.)

32лк

экзамен

Лекционный курс

Логика предикатов 1-го порядка. Интерпретация. Модель. Общезначимость и выполнимость.

Теорема о монотонной замене.

Теорема о дедукции.

Исчисление натуральных выводов. Непротиворечивость. Полнота.

Алгоритмическая неразрешимость проблемы выводимости.

Формальные системы типа арифметики. Теорема Геделя о неполноте и непротиворечивости.

Логические исчисления.

Возможные приложения в программировании.

ЯЗЫКИ И СИСТЕМЫ ПРОГРАММИРОВАНИЯ

(обяз. мин.)

32лк

экзамен

Лекционный курс

Проблема модульности.

Переносимость. Уровни и средства.

Средства отладки. Надежность.

Машинно-ориентированные языки.

Макропроцессоры.

Трансформации программ и смешанные вычисления.

Сравнительный анализ языков программирования.

Функциональные языки. Языки описания объектов.

Системы синтеза программ.

Языковое обеспечение взаимодействия человека с вычислительной машиной (меню, шаблоны, ключевые слова, использование диалога, окна). Ориентация на пользователя.

СТРУКТУРА ВЫЧИСЛИТЕЛЬНЫХ**СИСТЕМ (обяз. мин.)**

32лк + 16лб

зачет с оценкой

А. Лекционный курс

Основные типы вычислительных структур и их компоненты. Процессор. Память. Каналы. Межпроцессорное взаимодействие. Периферия. Диски. Ленты. Типы печатающих устройств. Графопостроители. Основные типы дисплеев. Контроллеры периферийных устройств.

Линии связи.

Элементная база вычислительных машин.

Пути повышения скорости обработки информации.

Поколения машин. Пути развития вычислительной техники.

Многопроцессорные машины (типа Эльбрус).

Конвейерные машины.

Матричные машины.

Машины с динамической архитектурой. Поточковые машины.

Аппаратная реализация синхронизации.

Номенклатура и технические характеристики отечественных вычислительных систем.

Б. Лабораторные занятия

Изучение простейших физических характеристик отдельных узлов вычислительных машин и интерфейсов между ними.

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ**ДАНЫХ (обяз. мин.)**

48лк + 32лб

зачет, экзамен

А. Лекционный курс

Общие понятия. Уровни абстракции.

Основные модели. Реляционная модель. Сетевая модель.

Иерархическая модель. Сравнение моделей.

Языки описания данных. Языки манипулирования данными.

Проектирование баз данных.**Защита данных.****Неоднородные и распределенные базы данных.**

Физическая организация баз данных. Индексированные файлы. Файлы с записями переменной длины. Поиск по ключевым полям. Поиск по частичному соответствию. Приемы оценки эффективности баз данных.

Характеристики распространенных отечественных и зарубежных систем управления базами данных.

Б. Работа за дисплеем

Освоение современных систем управления базами данных. Проектирование и создание учебных баз данных.

**ПРИБЛИЖЕННОЕ РЕШЕНИЕ
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**

32лк + 32лб

зачет с оценкой

А. Лекционный курс

Методы решения задачи Коши для обыкновенных дифференциальных уравнений.

Методы решения линейных и нелинейных краевых задач для обыкновенных дифференциальных уравнений.

Методы сеток.

Методы решения многомерных задач.

Методы решения интегральных уравнений.

Обзор современных программных систем для приближенного решения дифференциальных уравнений. Сочетание символьных и численных методов.

В. Работа за дисплеем

Освоение и использование программных систем для приближенного решения уравнений. Анализ применяемых программ. Самостоятельное программирование для конкретных задач приближенных вычислений.

ВЫЧИСЛИТЕЛЬНАЯ ГЕОМЕТРИЯ

32лк

экзамен

Лекционный курс

Проектирование кривых и поверхностей.

Конструирование с помощью поперечных сечений.

Распознавание пересечений. Выделение невидимых элементов изображений.

Способы представления геометрических объектов. Представление растров, представленные грамматиками. Основные структуры данных, используемые при решении геометрических задач. Алгоритмы решения простых геометрических задач (например, задачи о пересечении выпуклых многогранников, задачи о положении точки в данном разбиении плоскости, задачи о построении триангуляции, простейшие задачи о движении фигур).

МАШИННАЯ ГРАФИКА (обяз. мин.)

32лк + 32лб

зачет с оценкой

А. Лекционный курс

Графические устройства. Дисплей. Графопостроители. «Мышь». Планшеты. Матричные принтеры.

Организация дисплейного файла и файла изображения. Языковые конструкции машинной графики. Языки описания изображений.

Графические стандарты.

Номенклатура и технические характеристики отечественных графических устройств.

Б. Работа за дисплеем

Освоение графической периферии и программных систем машинной графики. Решение задач.

КУРСОВАЯ РАБОТА (завершение)

96лб

СПЕЦКУРСЫ (факультативно)

Общее время перечисленных занятий в 6-м семестре, не считая времени на курсовую работу и спецкурсы: всего 352 ч, в неделю 22 ч)

(Дисплейное время в 6-м семестре: всего 192 ч, в неделю 12 ч на одного студента)

7-й СЕМЕСТР (4-й курс)
ПРЕДСТАВЛЕНИЕ ЗНАНИЙ (обяз. мин.)

32лк + 32дс зачет, экзамен

- А. Лекционный курс**
Виды знаний. Декларативные и процедурные знания.
Языки искусственного интеллекта. Базы знаний.
Методы представления знаний. Сетевая и реляционная техника представления знаний. Фреймы. Системы продукций.
Анализ текста на естественном языке.
Приложения. Вопросно-ответные и экспертные системы. Обучающие системы.
- Б. Работа за дисплеем**
Освоение и модификация систем представления знаний.

МИКРОПРОЦЕССОРЫ (обяз. мин.)

48лк + 32дс + 16лб зачет, экзамен

- А. Лекционный курс**
Архитектура микропроцессоров. Управление и синхронизация. Системы команд. Способы адресации.
Программирование. Реализация основных операторов. Подпрограммы. Загрузчик. Ассемблер. Компиляторы.
Интерфейс. Порты ввода-вывода. Внешние устройства. Прерывания. Доступ к памяти. Секционированные микропроцессоры. Синхронизация микропроцессоров.
Примеры применений.
Номенклатура и технические характеристики отечественных микропроцессоров.
- Б. Занятия за дисплеем**
Освоение систем команд, языков ассемблера и операционных систем микропроцессоров.
- В. Лабораторные занятия**
Изучение основных физических интерфейсов микропроцессоров.

ЭКСТРЕМАЛЬНЫЕ ЗАДАЧИ

32лк + 16 дс зачет, экзамен

- А. Лекционный курс**
Выпуклая оптимизация. Задачи проектирования и анализа программных и вычислительных систем, приводящие к выпуклым экстремальным задачам. Геометрия выпуклых множеств.
Линейное и полиномиальное выпуклое программирование. Алгоритмы, оценки сложности, практические рекомендации.
Краткий обзор недифференцируемых и стохастических задач и методов их решения. Обзор современных программных систем для решения выпуклых экстремальных задач.
- Б. Работа за дисплеем**
Освоение программных систем решения экстремальных задач.

**УПРАВЛЕНИЕ ТЕХНИЧЕСКИМИ
СРЕДСТВАМИ (обяз. мин.)**

32лк + 32дс зачет с оценкой

- А. Лекционный курс**
Принципы управления реальными объектами и системами. Интерфейс с приборами. Типы обрабатываемой информации.
Технические средства автоматизации эксперимента. Оборудование для связи с объектом. Стандарты.
Языки моделирования. Представление объекта в языке высокого уровня (мониторы, задачи).
Проблема реального времени. Алгоритмические трудности. Требования к операционной системе.
Языки программирования систем реального времени. Примеры конкретных задач.
- Б. Работа за дисплеем**
Практическая работа по курсу. Освоение языков программирования и систем управления.

СПЕЦКУРСЫ И СПЕЦСЕМИНАРЫ
зачеты, экзамены

КУРСОВАЯ РАБОТА (начало)

80 дс

(Общее время перечисленных занятий в 7-м семестре, не считая времени на курсовую работу, спецкурсы и спецсеминары: всего 272 ч, в неделю 17 ч)
(Дисплейное время в 7-м семестре: всего 192 ч, в неделю 12 ч на одного студента)

8-й СЕМЕСТР (4-й курс)

ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

32лк + 32дс зачет, экзамен

- А. Лекционный курс**
Идеи логического вывода в программировании. Соответствие между логическими и программными конструкциями. Методы логического описания семантики языков программирования. Логическая верификация программ. Языки логического программирования.
Применение дедуктивных конструкций.
Построение программ по выводу.
Методы синтеза программ. Примеры систем синтеза программ. Приложения.
- Б. Занятия за дисплеем**
Освоение и модификация конкретных систем логического программирования.

**ТЕХНОЛОГИЯ И ЭКОНОМИКА
РАЗРАБОТКИ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ (обяз. мин.)**

32лк зачет с оценкой

- Лекционный курс**
Программа как продукт. Промышленные требования к программам. Стандарты. ЕСПД.
Организация разработки программных средств. Формы коллективных разработок. Документирование программ. Технологическая дисциплина разработки программных средств. Модификация ранее созданных программных систем.
Инструментальные средства поддержки программных разработок. Интегрированные технологические комплексы разработки программного обеспечения.
Разработка встроенного программного обеспечения технических средств.
Жизненный цикл программ. Обеспечение эксплуатации программ. Сопровождение. Организация управления и анализа жизненного цикла программ. Экономическая оценка отдельных его этапов и оценка в целом.
Юридические аспекты.

ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ (обяз. мин.)

48лк + 32дс зачет, экзамен

- А. Лекционный курс**
Структура вычислительных сетей. Топология. Каналы. Характеристики каналов и интерфейсов. Классификация сетей.
Коммутация каналов и пакетов.
Маршрутизация. Надежность.
Протоколы связи. Уровни. Рекомендации серии X.
Примеры сетей. Электронная почта.
Протоколы, обеспечивающие распределенные вычисления.
Методы разработки и анализа сетей. Защита информации в сетях.
Номенклатура отечественного сетевого оборудования и его технические характеристики.
- Б. Работа за дисплеем**
Освоение распределенных операционных систем и протоколов различных уровней и различного назначения.

ЭКСТРЕМАЛЬНЫЕ ЗАДАЧИ

32лк + 16дс

зачет, экзамен

А. Лекционный курс

Дискретные задачи оптимизации.

Транспортная задача. Поточковые задачи. Задача о назначениях. Поточковые задачи, возникающие при анализе вычислений и программ.

Задачи распределения работ (задачи календарного планирования). Приложение к проектированию и анализу вычислительных сетей, операционных, управляющих и других программных систем.

Методы сокращения перебора. Метод ветвей и границ. «Гриды» — алгоритмы и матроиды. Описание исходных данных грамматиками (например, графовыми) и применение методов синтаксического анализа. Другие эвристические приемы сокращения перебора.

Обзор современных программных систем для решения дискретных экстремальных задач.

Б. Работа за дисплеем

Освоение программных систем для решения дискретных экстремальных задач.

СТРУКТУРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ ПРОМЫШЛЕННОСТИ (обяз. мин.)

32лк

зачет с оценкой

Лекционный курс

Схематическая структура интегрированного производственного комплекса. Информационный и функциональный аспекты.

Уровни и средства автоматизации. Назначение отдельных систем.

Структура и примеры АСНИ, АСУ, САПР, АСПП, АСУТП, ГАЦ, АСИ. Технические характеристики оборудования.

Языковые средства, используемые в интегрированных производственных комплексах.

СПЕЦКУРСЫ И СПЕЦСЕМИНАРЫ

зачеты, экзамены

КУРСОВАЯ РАБОТА (завершение)

112дс

(Общее время перечисленных занятий в 8-м семестре, не считая времени на курсовую работу, спецкурсы и спецсеминары: всего 256 ч, в неделю 16 ч)

(Дисплейное время в 8-м семестре: всего 192 ч, в неделю 12 ч на одного студента)

9-й СЕМЕСТР (5-й курс)

ЭРГОНОМИКА И ПСИХОЛОГИЯ

32лк

зачет

Лекционный курс

Принципы и задачи эргономики. Основные эргономические характеристики человека. Эргономика и производительность труда.

Эргономические аспекты программных систем.

Элементы инженерной психологии. Вопросы психологии программирования.

СПЕЦСЕМИНАР ПО СИСТЕМНОМУ

АНАЛИЗУ (обяз. мин.)

32ч

зачет

Семинар проводится на базе конкретной промышленной разработки.

СПЕЦСЕМИНАР ПО СИСТЕМАМ АВТОМАТИЗАЦИИ В ПРОМЫШЛЕННОСТИ (обяз. мин.)

32 ч

зачет

Семинар знакомит студентов с актуальными задачами, стоящими перед промышленностью региона.

СПЕЦКУРСЫ И СПЕЦСЕМИНАРЫ

зачеты, экзамены

ДИПЛОМНАЯ РАБОТА (начало)

240дс

(Общее время перечисленных занятий в 9-м семестре, не считая времени на дипломную работу, спецкурсы и спецсеминары: всего 96 ч, в неделю 6 ч)

(Дисплейное время в 9-м семестре: всего 240 ч, в неделю 15 ч на одного студента)

10-й СЕМЕСТР (5-й курс)

ПРЕДДИПЛОМНАЯ ПРАКТИКА ДИПЛОМНАЯ РАБОТА (завершение)

288дс

(Дисплейное время в 10-м семестре: всего 288 ч, в неделю 18 ч на одного студента.)

СПЕЦКУРСЫ

(Буквы а, б, в или г, стоящие после названия спецкурса, указывают, для каких из перечисленных на стр. 3 специализаций предназначен данный курс.)

1. Переносимые операционные системы (а, б).
 2. Описание и моделирование языковых механизмов (а, б, в, г).
 3. Естественно-языковые системы (в, г).
 4. Диалоговые системы (в, г).
 5. Техника трансляции (а, б, в).
 6. Пакеты прикладных программ (б, в).
 7. Управление экспериментом с помощью вычислительных машин (б, в).
 8. Алгоритмы информационного поиска (а, б, в, г).
 9. Алгоритмы анализа сигналов (б).
 10. Моделирование и обработка изображений (б, в).
 11. Распознавание образов (в).
 12. Машинное зрение (б).
 13. Управление роботами (б).
 14. Спецпроцессоры на СБИС (а, б, в).
 15. Теория сложности вычислений (б, в, г).
 16. Теория логического вывода (б, в, г).
 17. Теория схем программ (а).
 18. Теория кодирования (б).
 19. Символьные вычисления (в).
 20. Информационно-поисковые системы (в, г).
 21. Семантика языков программирования (а, в).
 22. Автоматическое доказательство теорем (в).
 23. Теория массового обслуживания (а, б).
 24. Имитационное моделирование (а, б, в, г).
 25. Автоматизированные обучающие системы (а, б, в, г).
- Спецкурсы по конкретным языкам программирования

СПЕЦСЕМИНАРЫ

- Спецсеминары по изучению конкретных программных систем
- Спецсеминары по изучению конкретной алгоритмики
- Новые направления в языках программирования
- Спецсеминары по разработке и реализации учебных систем

Статья поступила 27 мая 1985 г.

В. Б. Бетелин

ПРОБЛЕМА АВТОМАТИЗАЦИИ ПОДГОТОВКИ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

В процессе разработки систем программного обеспечения одним из наиболее трудоемких этапов является создание проектной и эксплуатационной документации в соответствии с требованиями ЕСПД. Эта документация представляет собой совокупность взаимосвязанных документов, таких, например, как «цели и масштабы системы», «предварительный внешний проект», «детальный внешний проект», «руководство пользователя» и т. д. [1]. Не менее трудоемкой задачей на стадии сопровождения программной системы являются оперативная подготовка и выпуск документации, которые соответствуют новым версиям системы. Таким образом, повышение производительности труда программиста в значительной мере связано с автоматизацией обработки текстовой документации.

В настоящее время наиболее распространенным средством автоматизации этой деятельности являются универсальные экранные текстовые редакторы, применение которых ограничивается, в основном, подготовкой программной документации. Как показывает имеющийся опыт, использование универсальных редакторов для этих целей требует от программиста выполнения значительного объема рутинных действий, т. е. малоэффективно. Причина заключается прежде всего в примитивизме основных объектов универсальных редакторов (литера, строка литер) и операций над этими объектами. Кроме того, явно недостаточен объем сервисных возможностей (форматирование, выравнивание, поиск и т. д.), специфических для систем обработки текстов [2].

Необходимым условием автоматизации работ, связанных с обработкой текстов, является создание комплексной системы управления документацией, позволяющей хранить документы в единой базе данных проекта (например, программной системы), организовывать целенные воздействия на отдельные документы и на всю их совокупность, поддерживать и контролировать связи между отдельными документами и их фрагментами, обеспечивать соответствие версий различных документов. Важной компонентой такой системы является специализированный редактор, ориентированный на создание текстовой документации, в число основных объектов которого включены слова, предложения, разделы, главы, оглавления. Создание такого редактора в первую очередь связано с разработкой более сложной концептуальной модели документа, чем та, на основе которой реализовано большинство существующих универсальных редакторов.

В настоящей работе предлагается такая модель документа, приводится краткое описание функциональных возможностей конкретного редактора, реализованного на основе рассмотренной модели. Редактор внедрен в эксплуатацию в научно-производственном центре АН СССР и ПО ЗИЛ по проблемам САПР в машиностроении.

Концептуальная модель документа

Логически любой документ представляет собой упорядоченную последовательность разделов (глав, параграфов и т. д.), каждый из которых может быть представлен совокупностью подразделов. Подразделы тоже могут иметь сложную структуру. Принципиальных ограничений на число уровней вложенности разделов в пределах одного документа, вообще говоря, не существует, однако на практике, как правило, используется

не более трех уровней. Разделы последнего уровня представляют собой последовательность абзацев, каждый из которых включает одно или несколько предложений. Предложение есть последовательность слов, а слово — последовательность литер.

Таким образом, естественная модель документа — это структура типа дерево, терминальными узлами которой являются литеры, а нетерминалами — главы, разделы, абзацы, предложения и слова. Минимальный набор операций над этими объектами, необходимый для создания и изменения экземпляра такой модели, включает операции создания, уничтожения, изменения, копирования, перемещения и поиска.

Конечной целью процесса создания документа является его печатное представление, получаемое путем отображения структурных объектов документа вместе с некоторой служебной информацией на последовательность пронумерованных страниц одинакового размера. Чтобы получить печатное представление документа, необходимо для каждой структурной единицы задать формат ее размещения. Например, для каждой главы надо определить такие параметры, как положение заголовка и первого абзаца, требуется ли выровнять текст по левому (правому) краю или центру, должна ли начинаться глава с новой страницы и т. д.

Кроме формата, для всех структурных объектов необходимо определить параметры их визуализации на печатающем устройстве или видеотерминале. К числу таких параметров относятся тип шрифта (прописные, строчные буквы, курсив и т. д.), подчеркивание, цвет, позитив (негатив), чувствительность к световой указке, мерцание и т. д. Например, заголовки разделов можно выделить, используя для их изображения большие буквы с подчеркиванием. Таким образом естественно дополнить модель документа, связан с каждым типом структурного объекта некоторое множество атрибутов визуализации и форматирования.

Неотъемлемой частью документа являются такие служебные разделы, как оглавление, предметный указатель, список литературы. Оглавление достаточно компактно представляет верхние уровни древовидной структуры документа и обеспечивает практически прямой доступ (с точностью до подраздела) к требуемому фрагменту текста документа. Аналогичным образом используется предметный указатель. Все перечисленные объекты также необходимо ввести в состав модели документа. Кроме того, модель дополняется объектом типа словарь, который можно создать для документа, главы и раздела.

Краткое описание функциональных возможностей редактора документации

На основе предложенной модели спроектирован и реализован специализированный редактор текстовой документации.

Обобщенный курсор. В основном режиме редактор обеспечивает возможность работы с такими объектами, как литера, слово, предложение, абзац, раздел. В любой момент времени существует только один объект какого-либо из перечисленных типов, над которым можно выполнять операции удаления, перемещения, копирования и т. д. Этот объект называется текущим объектом и изображается на экране видеотерминала в ви-

верном виде. Передвижение обобщенного курсора производится следующими функциональными клавишами:

→ (←) — текущим становится следующий (предыдущий) объект того же типа;

↑ — текущим становится объект, обьсмлющий данный; если текущий объект — символ, то в результате выполнения этой операции текущим объектом станет слово, которое его содержит; для слова результатом будет содержащее его предложение; для предложения — абзац; для абзаца — раздел и т. д.;

↓ — текущим становится первый подобъект текущего объекта; если текущий объект — раздел, то результатом выполнения операции будет первый подраздел или первый абзац; для абзаца результат — первое его предложение; для предложения — первое слово и т. д.

Над текущим объектом определены следующие операции: удалить, переместить влево (вправо), поменять местами с предыдущим (последующим), создать перед текущим объектом объект того же типа, изменить атрибуты объекта.

Пример. Пусть на экране видеотерминала визуализируется следующий абзац, состоящий из одного предложения.

ПРИМЕР РАБОТЫ СО СПЕЦИАЛИЗИРОВАННЫМ РЕДАКТОРОМ ДОКУМЕНТАЦИИ

Текущим объектом является буква «Р» в слове «РАБОТЫ», она выделена в тексте инверсией. Если в этот момент нажать клавишу ↑, то текущим объектом станет слово «РАБОТЫ», что отобразится на экране следующим образом.

ПРИМЕР РАБОТЫ СО СПЕЦИАЛИЗИРОВАННЫМ РЕДАКТОРОМ ДОКУМЕНТАЦИИ

Если далее нажать клавишу →, то текущим объектом станет слово, следующее за словом «РАБОТЫ», т. е. предлог «СО». Повторное нажатие этой клавиши сделает текущим слово «СПЕЦИАЛИЗИРОВАННЫМ», что на экране видеотерминала отобразится следующим образом.

ПРИМЕР РАБОТЫ СО СПЕЦИАЛИЗИРОВАННЫМ РЕДАКТОРОМ ДОКУМЕНТАЦИИ

Если теперь нажать клавишу ↓, то текущим объектом станет буква «С» в слове «СПЕЦИАЛИЗИРОВАННЫМ».

Атрибуты объектов. В первой версии редактора множество возможных атрибутов объектов включает атрибуты форматирования и визуализации. Атрибуты форматирования могут задаваться для документа в целом, раздела и абзаца. Допускаются следующие значения атрибутов: выравнивать текст по левой и правой границам, центрировать текст, не изменять текст.

Значение атрибута *выравнивать* означает, что в момент его присваивания и далее при каждом изменении текста абзаца (вставка, удаление символов, слов, предложений) последний будет автоматически выравниваться по левой и правой границам, положения которых определяются программистом непосредственным указанием на экране видеотерминала.

Значение атрибута *центрировать* означает, что текст абзаца при каждом его изменении будет автоматически выравниваться относительно некоторой центральной точки, определяемой путем прямого указания на экране видеотерминала. Это значение атрибута форматирования наиболее характерно для заголовков документов и разделов, которые относятся к объектам типа абзац.

В ряде случаев смысл отдельных фрагментов документа может зависеть от взаимного расположения строк и символов внутри строк. К ним относятся таблицы, псевдографические рисунки, схемы и т. д. Например, фрагмент

X						
0.1	1.6					
0.2	2.8					
0.3	7.9					

не эквивалентен фрагменту

X	0.1	1.6	0.2	2.8	0.3	7.9
---	-----	-----	-----	-----	-----	-----

Для того чтобы обеспечить возможность работы с такими фрагментами, в модель документа введен специальный тип объектов — *неструктурированная последовательность строк*. Создание и изменение объектов этого типа ведется в режиме традиционного экранного редактирования последовательности символьных строк. Поскольку фрагменты этого типа не имеет смысла автоматически выравнивать, для них введено специальное значение атрибута форматирования — *не выравнивать*.

Атрибуты визуализации могут быть заданы для любого объекта документа. Множество допустимых значений включает: прописные и строчные буквы, курсив, подчеркивание, негатив (позитив), мерцание. Значения атрибутов любого объекта определяются в момент его создания либо явно пользователем, либо по умолчанию редактором.

Разделы и оглавление. Каждый раздел (подраздел) документа имеет следующую структуру:

номер — специальный объект, который автоматически генерируется редактором, например 1.2.4.;

заголовок — объект типа абзац;

текст — последовательность объектов типа подраздел или абзац.

Документ в целом представляет собой раздел самого высокого уровня. Относительно уровня текущего раздела документа можно создать раздел того же уровня (клавиша Р=), более высокого уровня (клавиша Р↑), более низкого уровня (клавиша Р↓).

Процесс создания нового документа начинается с создания первого раздела, для чего необходимо нажать клавишу Р=. Далее вводятся заголовок (первый введенный абзац считается заголовком), затем текст раздела. Создание первого подраздела осуществляется с помощью клавиши Р↓, а всех последующих того же уровня с помощью клавиши Р=. Вложенность подразделов принципиально не ограничена.

При создании и модификации заголовков, при перемещении разделов происходит автоматическая коррекция оглавления, которое создается редактором для любого документа, включающего хотя бы один раздел. Оглавление можно редактировать, что влечет за собой изменение заголовков в тексте документа, а также создание новых разделов и перемещение уже имеющихся. В любой момент времени при работе с оглавлением определен его текущий элемент, соответствующий одному из разделов (подразделов) документа. Этот элемент выделяется в тексте оглавления обобщенным курсором и является аргументом всех операций, определяемых на уровне оглавления. Например, в результате выполнения операции. Текст раздела станет доступным для изменения текста того раздела, на который указывал в оглавлении обобщенный курсор. Если далее выполнить операцию Оглавление, то произойдет возврат на уровень оглавления, причем текущим объектом станет элемент оглавления, соответствующий тому разделу документа, из которого осуществлен этот переход.

Групповые операции над объектами. Редактор документов обеспечивает возможность определения некоторых операций редактирования сразу для группы объектов (символов, слов, предложений, абзацев, разделов). Группы объектов определяются с помощью операции маркирования и выделяются на экране видеотерминала пониженной яркостью. Групповые операции используются, как правило, для копирования и перемещения фрагментов текста в пределах одного или нескольких документов. В число групповых операций входят: удаление, копирование, перемещение (выделенная группа помещается перед текущим объектом, исходная копия в тексте уничтожается), изменение атрибутов, расформирование группы.

Средства откатки. Процесс редактирования документа представляет собой последовательное выполнение собственно операций редактирования, а также вспомогательных операций типа позиционирования курсора, изменения атрибутов объектов и т. д. Выполнение каждой такой операции изменяет состояние документа: из-

меняется текст какого-либо абзаца, становится текущим другой объект или изменяется режим визуализации некоторого слова и т. д. Другими словами, любой последовательности операций соответствует последовательность состояний документа. С помощью операции Откатить можно двигаться назад по этой цепочке состояний на любую ее глубину, вплоть до состояния на момент его создания. Эта операция является аннулирующей для любой операции редактора (основной и вспомогательной), кроме себя самой, т. е. откатить результат операции Откатить с помощью самой этой операции невозможно. Для этой цели служит операция Отменить, которая аннулирует результат любой предшествующей операции редактора, включая операцию Откатить и собственно саму операцию Отменить. Так, например, последовательность операций Отменить Отменить и Откатить Отменить эквивалентны в том смысле, что после их выполнения состояние документа не изменится.

Специальная пара операций редактора позволяет осуществлять откатку выделенной (маркированной) группы объектов документа. Операция Откатить группу позволяет восстанавливать любое предыдущее состояние объектов группы независимо от состояния других объектов документа. Вторая операция позволяет двигаться вперед по цепочке состояний. Это имеет смысл только в том случае, если объект после восстановления не изменялся.

Некоторые сведения о реализации. Специализированный редактор документации реализован на современ-

ной мини-ЭВМ в рамках экспериментальной системы автоматизации труда программиста, включающей пре-процессор на языке Фортран с языка типа псевдокод и экраный редактор, ориентированный на этот язык. Объем реализации составляет более 30 тыс. строк на языке Фортран-77.

Существенно, что около 50 % общего объема программных текстов редактора документации были полностью или с небольшими изменениями заимствованы из реализации редактора псевдокодных программ. К числу заимствованных модулей относится, например, модуль «виртуальный терминал», который обеспечивает надстроенным над ним прикладным системам независимость от типа физического терминала, оптимизацию операций вывода на экран, возможность перераспределения функций между клавишами на клавиатуре и т. д.

В настоящее время редактор внедрен в эксплуатацию в Научно-производственном центре АН СССР и ПО ЗИЛ. По предварительным оценкам производительность труда конструкторов, технологов и программистов при подготовке документации возросла в 6...8 раз.

ЛИТЕРАТУРА

1. Майерс Э. Надежность программного обеспечения.— М.: Мир, 1980.
2. Борковский А. Б., Хельбиг Г. Системы подготовки текста. Отчет рабочей группы— 18 КНВВТ.— М.: ВИНТИ, 1984, с. 73—87.

Статья поступила 2 сентября 1985 г.

УДК 681.03.85

Н. Н. Щелкунов, А. П. Дианов

ПРОГРАММИРОВАНИЕ МИКРОСИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Особенности микросистем (МС) реального времени (РВ) требуют развития соответствующих методов их программирования. В основе этих методов лежит многозадачный принцип организации систем. Эффективным средством программирования МС РВ являются специальные функционально-законченные группы управляющих процедур и наборы данных к ним — исполнительные ядра. Встраиваемые в аппаратуру прикладных систем стандартные исполнительные ядра составляют основу кремнивого программного обеспечения.

Особенности систем реального времени. МС РВ осуществляют оперативное взаимодействие с внешней средой. Они должны быть постоянно готовыми к приему входной информации и обязаны обеспечить соответствующую реакцию за достаточно короткое время отклика. Малое время реакции и параллельная приоритетная обработка являются отличительными чертами МС РВ. При разработке программного обеспечения необходимо учитывать сложные временные и приоритетные взаимоотношения между отдельными функциями МС РВ.

Многозадачная организация программ. В основу современной техники программирования МС РВ положен опыт организации мультипрограммной обработки [1—3]. Параллельное выполнение нескольких задач легко моделируется, если они полностью независимы. В действительности задачи взаимодействуют между собой, синхронизируют свою работу, вызывают друг друга, обмениваются сообщениями и данными, разделяют основные ресурсы микросистемы: процессорное время, память, каналы ввода-вывода. В борьбе за ресурсы они вступают в противоречия, порождая трудно разрешимые конфликтные ситуации. В процессе работы задачи создаются, выполняются, временно приостанавливаются, возобновляются и уничтожаются.

Реализация ряда этих фундаментальных функций, присущих многозадачному режиму работы, вознаграждается

на так называемое исполнительное ядро РВ, составляющее центральную часть программного обеспечения МС РВ. Исполнительное ядро предусматривает:

- инициализацию системы и динамическое управление ее составом;
- распределение времени микропроцессора (планирование задач) и других ресурсов МС;
- организацию и синхронизацию межзадачного взаимодействия;
- связь с системой прерываний микропроцессора.

Задачи могут рассматриваться независимо друг от друга, если их взаимодействие реализуется не непосредственно, а через такое ядро, играющее роль межзадачного интерфейса и разрешающее все конфликтные ситуации. Обращение к ядру выполняется с помощью строго определенного набора системных вызовов, которые можно рассматривать как расширение множества инструкций микропроцессора. Системные вызовы ядра называют примитивами, отображая тот факт, что они, так же как и команды, являются неделимыми программными единицами, однако в отличие от последних представляют собой некоторые подпрограммы. Неделимость означает гарантию исключительно индивидуального доступа к набору данных, с которыми работает примитив в течение всего цикла исполнения.

Примитивы оперируют со специальными управляющими структурами данных — объектами, важнейшими среди которых являются объекты-задачи, объекты-сообщения и объекты-обменники. Эти типы объектов используются для представления состава и состояния МС РВ.

Задачи и их состояния. При работе системы ядро поддерживает ряд управляющих структур и списков. Одной из таких структур является дескриптор системы SYS: SYSTEM, содержащий указатели основных системных списков. Дескриптор создается в области ОЗУ при

инициализации системы и может иметь следующую структуру:

```
Type SYSTEM is
  record
    FIRST_TASK:TASK      -- указатели списка
    LAST_TASK:TASK      -- TASK_LIST
    FIRST_READY:TASK     -- указатели списка
    LAST_READY:TASK     -- READY_LIST
    FIRST_SUSP:TASK      -- указатели списка
    LAST_SUSP:TASK      -- SUSP_LIST
    FIRST_DELAY:TASK     -- указатели списка
    LAST_DELAY:TASK     -- DELAY_LIST
    FIRST_EXCH:EXCHANGE -- указатели списка
    LAST_EXCH:EXCHANGE  -- EXCH_LIST
  end record;
```

Каждая задача в МС описывается специальным объектом — дескриптором задачи. Последний представляет собой упорядоченный набор данных, содержащих основную информацию о состоянии задачи и ее полномочиях, необходимую для ядра. Пример простого дескриптора задачи:

```
Type TASK is access
  record
    NAME:IDENTIFICATOR -- имя задачи
    PRIORITY:INTEGER    -- приоритет
    STACK_PTR:POINTER  -- указатель стека
    STATUS:CHARACTER    -- состояние задачи
    DELAY:INTEGER       -- время ожидания
    EXCH_WAIT:EXCHANGE -- обменник ожидания
    NEXT_DELAY:TASK     -- указатель DELAY_LIST
    NEXT:TASK           -- указатель других списков
    NEXT_TASK:TASK      -- указатель TASK_LIST
  end record;
```

Конкретная структура и состав дескриптора, как и других объектов системного типа, зависит от реализации ядра. В приведенном примере дескриптор задачи содержит девять полей. Три последних поля используются в качестве ссылок на следующие дескрипторы, составляющие различные системные списки и очереди. Так, все включенные в систему SYS задачи вносятся в общий список задач TASK_LIST, указателями начала и конца которого служат соответственно поля FIRST_TASK и LAST_TASK дескриптора системы. Указателем следующего элемента в списке является поле NEXT_TASK дескриптора задачи.

Каждая задача может находиться в одном из трех основных состояний: активности ACTIVE, готовности READY и ожидания WAIT кодируемых полем STATUS.

Часто различают еще одно вспомогательное состояние задачи: состояние приостанова или подвешивания SUSP. Приостановленная задача хотя и присутствует в системе, однако не участвует в борьбе за ее ресурсы.

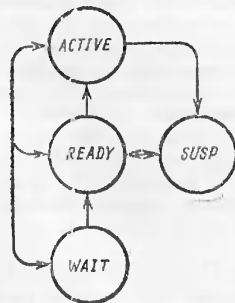


Рис. 1. Типовая диаграмма смены состояний

Приостановка позволяет упростить активную часть системы, что бывает полезно при отладке и для увеличения быстродействия. Некоторые ядра, например iRMX-86 [6], допускают многократное подвешивание задач. Типовая диаграмма переходов задачи из одного состояния в другое приведена на рис. 1. Отметим, что в состоянии приостанова может быть переведена только готовая или активная задача.

Планирование задач. Базовой функцией МС РВ является распределение процессорного времени задачам из числа готовых (планирование задач), которая возлагается на планировщика задач SCHEDULER. Эффективность системы во многом определяется алгоритмом планирования.

Планировщик обычно поддерживает список готовых задач READY_LIST, который упорядочивается согласно стратегии планирования. В качестве активной всегда выбирается задача, стоящая во главе списка. Указатель SYS.FIRST_READY начала очереди служит одновременно указателем активной задачи. Для ссылки на следующий элемент очереди готовых задач можно использовать поле NEXT дескриптора задачи. Трехэлементный список READY_LIST системы, состоящий из шести задач, приведен на рис. 2.

Для различных классов систем требуются свои стратегии планирования, в основе которых лежат два основных способа упорядочивания списка готовых задач: временной по принципу FIFO (первый пришел — первый будет обслужен) и приоритетный.

Учет времени выполнения задачи автоматически осуществляется в алгоритме циклического планирования типа round-robin (RR). Этот метод состоит в циклическом предоставлении равных квантов времени всем готовым задачам. Он используется в МС с разделением времени, когда предпочтение отдается более коротким задачам.

В общем случае величина кванта времени t_0 определяется типом, важностью или срочностью задачи, которые описываются с помощью приоритета задачи. Чем выше приоритет, тем длиннее квант времени, выделяемый данной задаче. Задача, истратившая свой квант времени, возвращается в конец очереди. Приоритет обычно кодируется в специальном поле PRIORITY дескриптора задачи.

Вторая группа методов планирования использует списки задач, упорядоченные по приоритетному принципу. Во главе списка всегда находится задача с наивысшим приоритетом. Задачи одинакового уровня приоритета записываются в список по правилу FIFO. Наиболее простым методом управления этой группой является приоритетное планирование. Процессор отдается задаче с наивысшим приоритетом, которая остается активной до достижения ею точки блокировки.

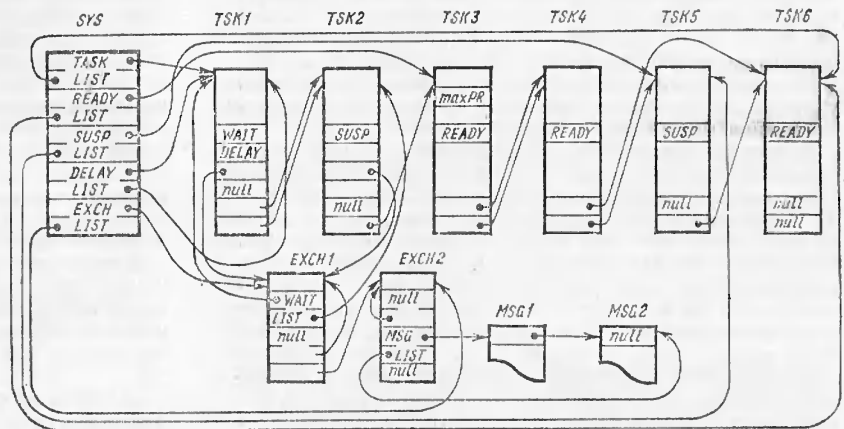
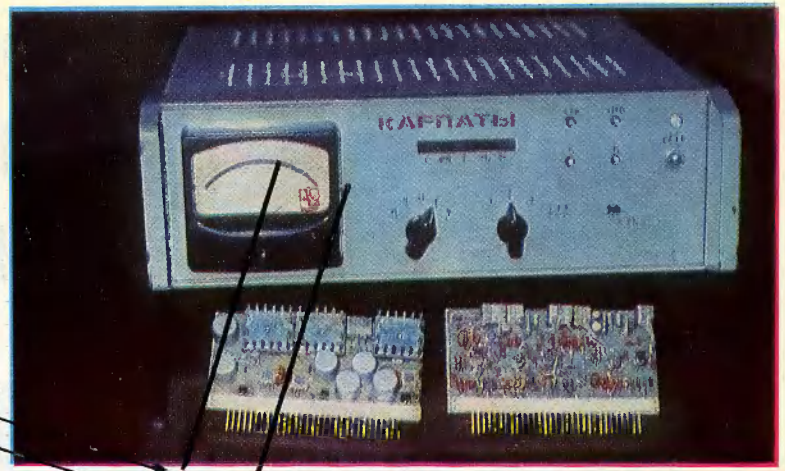


Рис. 2. Организация списков шестизадачной системы



Комплекс для управления процессом бурения нефтяных и газовых скважин

(К ст. Г. Н. Семенцова)



ИНТЕГРИРОВАННАЯ МАШИНОСТРОИТЕЛЬНАЯ СИСТЕМА КАПРИ

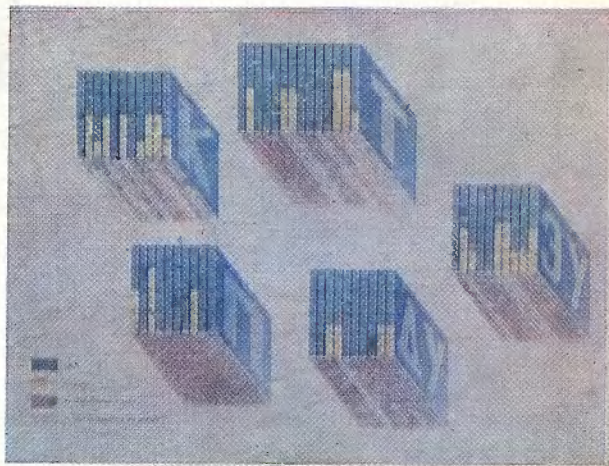


Интегрированная машиностроительная система (ИМС) КАПРИ обеспечивает комплексную автоматизацию проектно-конструкторских работ, технологической подготовки производства и изготовления экспериментальных изделий машиностроительного профиля в НИИ и КБ.

Основные автоматизированные функции системы КАПРИ:

конструирование типовых узлов, типовых и оригинальных деталей, включая информационное обеспечение процесса конструирования;

проектирование технологии изготовления деталей, получение управляющих программ для станков с числовым программным управлением (ЧПУ) и промышлен-



Структура интегрированной машиностроительной системы КАПРИ

Соотношение рутинных и творческих работ и возможности сокращения рутинных работ посредством автоматизации конструкторской, технологической и управленческой деятельности

Робототехнический модуль с роботом РК-1

ленных роботов, а также имитационная проверка качества этих программ;

организационно-экономическое управление производством, в том числе технико-экономическое и оперативное планирование, учет производства и его материально-технического обеспечения;

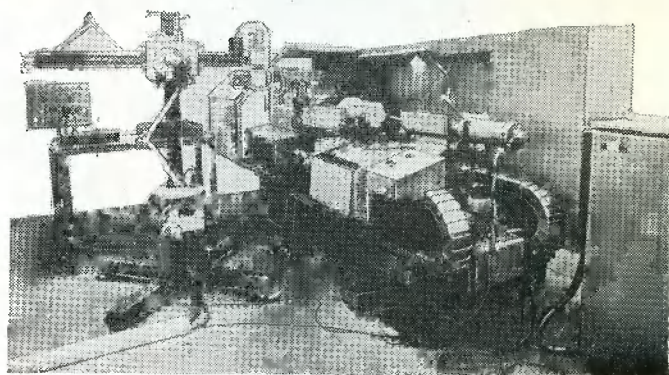
оперативно-диспетчерское управление, в том числе сменно-суточное планирование, контроль и анализ состояния оборудования и выполнения сменных заданий, оперативная коррекция заданий;

управление технологическими процессами на программно-управляемом технологическом и транспортно-складском оборудовании с применением промышленных роботов.

Таким образом, в процессе конструирования, проектирования технологии, производства и управления посредством автоматизации устраняется значительная часть рутинных нетворческих операций.

Автоматизированный производственный комплекс содержит 25 одно- и многооперационных токарных и фрезерно-сверлильно-расточных станков с микропроцессорными системами ЧПУ. Часть станков оснащена промышленными роботами.

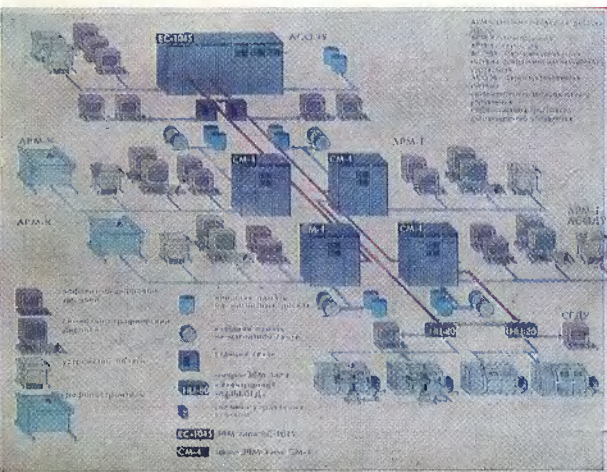
Успешно работает роботизированный комплекс, состоящий из токарного станка 16К20Т1 и промышленного робота РК-1. Система управления робота, выполнен-



ная на базе микроЭВМ «Электроника 60», обладает повышенной гибкостью и, вместе с тем, простотой настройки на новые операции.

ИМС КАПРИ реализована на комплексе средств вычислительной техники, в состав которого входят ЭВМ ЕС-1045, четыре АРМ-М на базе ЭВМ СМ-4, шесть автоматизированных рабочих мест конструкторов на базе символьно-графических станций «Графит», более 40 автоматизированных рабочих мест технологов, нормировщиков, начальников участков, мастеров и другого производственного персонала на базе алфавитно-цифровых дисплеев 15ИЭ-00-013. Управление технологическим оборудованием осуществляется встроенными в него микроЭВМ «Электроника 60», «Электроника НЦ-31» и станциями группового диспетчерского управления на базе микроЭВМ «Электроника МС 1201.1» и дисплеев 15ИЭ-00-013.

ЭВМ и терминалы включены как абоненты в локальную сеть связи.



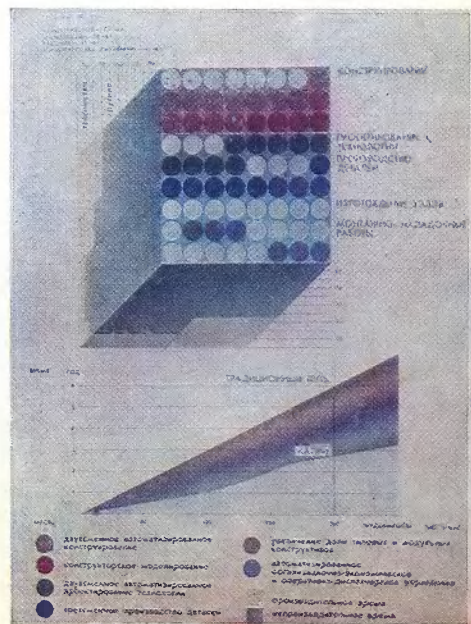
Участок станков с ЧПУ системы КАПРИ

Структура комплекса средств вычислительной техники системы КАПРИ

Сокращение сроков создания экспериментального стенда посредством автоматизации проектирования и производства

Все подсистемы ИМС КАПРИ работают в информационной взаимосвязи на основе центральной и региональной реляционных баз данных.

Внедрение системы КАПРИ дает возможность повысить производительность труда конструкторов в 1,5..3 раза, технологов в 2..4 раза, производственного персонала в 1,5..3 раза и тем самым сократить сроки создания крупных экспериментальных стендов в 1,5..2 раза. Существенно повышается качество и творческое содержание работ. Улучшается социальная структура коллективов инженерного обеспечения научных исследований в пользу высококвалифицированных творческих профессий. Снижается потребность в работниках дефицитных профессий: конструкторах, технологах, рабочих-станочниках и вспомогательных рабочих.

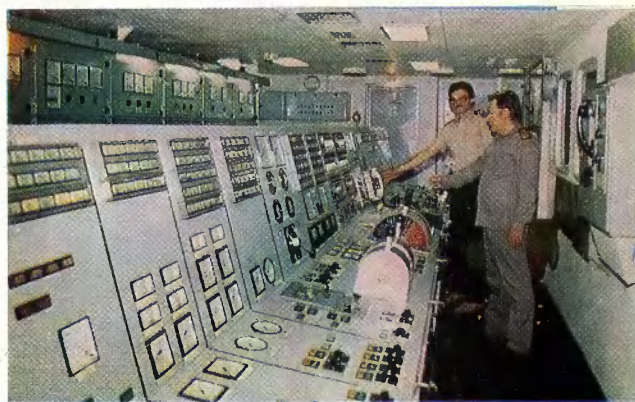
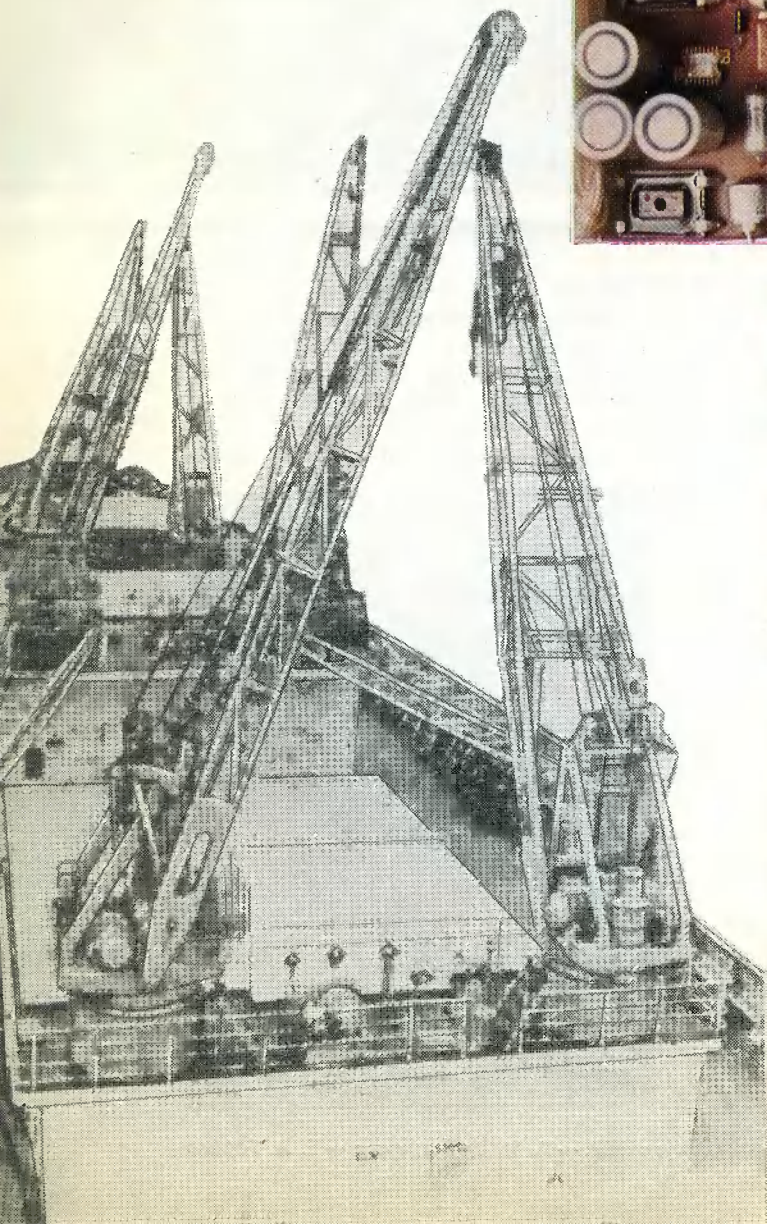
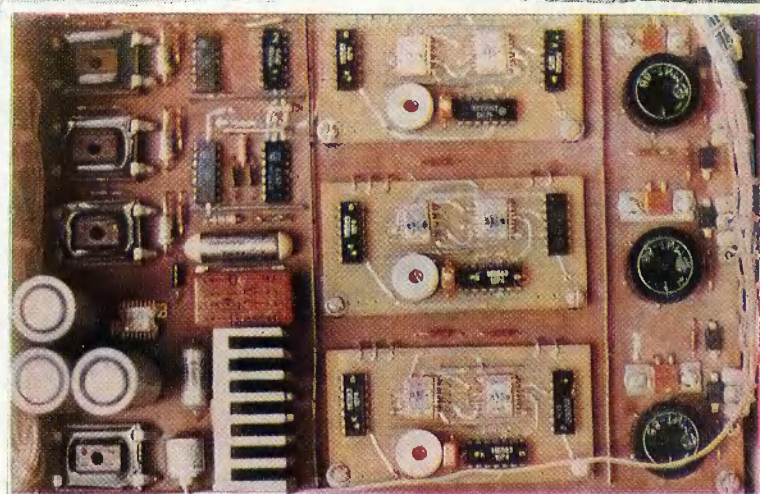
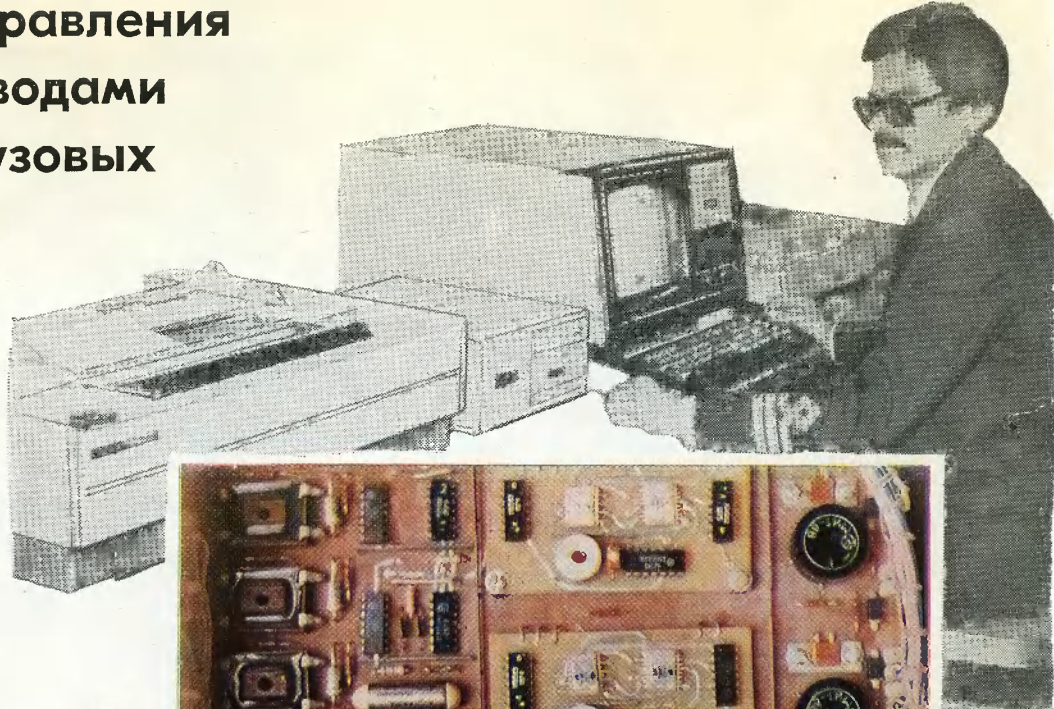


Художник — А. Н. Цибин, фото — Е. А. Горина

Система управления электроприводами судовых грузовых комплексов

МикроЭВМ,
судовая
электростанция
и грузовые
комплексы
объединены
в единую
систему

(К ст. С. А. Михайлова)



В MC PB чаще используется другой метод управления — приоритетное планирование с вытеснением. Согласно алгоритму готовая задача находится в состоянии выполнения до тех пор, пока она не достигнет точки блокировки или не появится новая готовая задача с более высоким приоритетом. Эта стратегия гарантирует, что активная задача будет всегда наиболее приоритетной из числа готовых. Она используется в MC, где невыполнение более приоритетной задачи приводит к серьезным последствиям.

Возможность динамического изменения приоритета задачи позволяет управлять временем ее исполнения. При этом приоритетное планирование с вытеснением допускает организацию различных вторичных стратегий управления задачами. Эти вторичные планировщики оформляются в виде запускаемой от таймера специальной высокоприоритетной задачи, изменяющей приоритеты всех других готовых задач.

Приоритетное планирование с вытеснением используется в кремниевых ядрах 80130 и VRTX [4], исполнительном ядре БРС/МОС PB CM-1800 [5].

Примитивы планирования. При обращении к ядру все регистры микропроцессора обычно заносятся в стек задачи, а содержимое указателя стека SP — в специальное поле STACK_PTR дескриптора текущей задачи. При выходе из ядра сначала восстанавливается указатель SP, который извлекается из дескриптора первой в списке READY_LIST задачи, а затем все остальные регистры, включая программный счетчик PC. Эти процедуры сохранения и восстановления текущего контекста будем обозначать как SAVE_CPU и LOAD_CPU соответственно.

Любой вызов ядра имеет вид:

```
procedure RQ_NAME is
begin
  SAVE_CPU;    -- сохранение контекста
               тело вызова;
  LOAD_CPU;    -- восстановление контекста
end RQ_NAME;  -- ста
```

В теле вызова стек задачи и регистры используются для собственных нужд ядра. Через них осуществляется передача и адресация параметров вызова ядра, а также результата его исполнения. При исполнении тела вызова могут быть изменены дескрипторы, контекст, а также адресуемые им данные других задач, что используется при организации их взаимодействия. Указателем текущей задачи служит ячейка SYS.FIRST_READY. При переходе от одной задачи к другой замена контекстов происходит автоматически при входе и выходе из ядра.

Вызов планировщика всегда связан с примитивами изменения состава и порядка очереди готовых задач. Среди таких примитивов прежде всего следует отметить следующие.

Примитив создания

```
procedure RQ_CREATE_TASK(TSK:TASK)
begin
  SAVE_CPU;    -- запись TSK в
  ENTER_TASK(TSK);    -- TASK_LIST
  ENTER_READY_TASK(TSK); -- запись TSK в
  LOAD_CPU;    -- READY_LIST
end RQ_CREATE_TASK;
```

Процедура включения задачи в список готовых выполняется в соответствии со стратегией планирования.

Примитив исключения

```
procedure RQ_DELETE_TASK(TSK:TASK)
begin
  SAVE_CPU;    -- удаление TSK
  DELETE_READY_TASK(TSK); -- из READY_LIST
  DELETE_TASK(TSK);    -- удаление TSK
  LOAD_CPU;    -- из TASK_LIST
end RQ_DELETE_TASK;
```

Примитив подвешивания

```
procedure RQ_SUSPEND_TASK(TSK:TASK)
begin
  SAVE_CPU;    -- удаление TSK
  DELETE_READY_TASK(TSK); -- из READY_LIST
  ENTER_SUSP_TASK(TSK); -- запись TSK в
  LOAD_CPU;    -- SUSP_LIST
end RQ_SUSPEND_TASK;
```

Для учета подвешенных задач в системе создается специальный список SUSP_LIST, указателями начала и конца которого служат ячейки SYS.FIRST_SUSP и SYS.LAST_SUSP соответственно. В качестве указателя следующего элемента списка используется свободное поле NEXT дескриптора задачи (см. рис. 2).

Примитив возобновления подвешенной задачи

```
procedure RQ_RESUME_TASK(TSK:TASK)
begin
  SAVE_CPU;
  DELETE_SUSP_TASK(TSK); --удаление TSK
                           -- из SUSP_LIST
  ENTER_READY_TASK(TSK); --запись TSK в
                           -- READY_LIST
  LOAD_CPU;
end RQ_RESUME_TASK;
```

Все эти примитивы работают со списком готовых задач. Возможны их модификации, например создание не только готовых, но и подвешенных задач, уничтожение любых задач, а также многократное подвешивание. В последнем случае в дескриптор задачи должен быть введен счетчик подвешивания.

Важное значение имеет примитив установки приоритета задачи

```
procedure RQ_SET_PRIORITY(TSK:TASK, PRIOR:INTEG)
begin
  SAVE_CPU;    --установка при-
               --оритета
  SET_PRIORITY(TSK, PRIOR); --указанной за-
                           --дачи из списка
  LOAD_CPU;    --TASK_LIST
end RQ_SET_PRIORITY;
```

Примитив может иметь доступ к любому элементу из общего списка TASK_LIST.

Эти процедуры, как и другие элементы ядра, требуют эффективного кодирования. Понимая важность такого кодирования, фирма U.S. Software предлагает базовый планировщик МП 68000, объектный код которого имеет размер менее 100B [6].

Сообщения и обменники. Задачи передают данные и служебную информацию с помощью сообщений — упорядоченных наборов данных, которые одна задача посылает другой. Сообщения — структурированные объекты системного типа, возможно переменной длины. Например:

```
type MESSAGE is access
record
  NEXT:MESSAGE;    -- указатель
                  --MSG_LIST
  RESPONSE:EXCHANGE; --обменник от-
                  --вета
  SIZE:INTEGER;    --размер бу-
                  --фера
  BUFFER:array(1..SIZE) of BYTE; --буфер данных
end record;
```

В сообщении может содержаться информация о его типе, важности, коды отправителя и получателя и т. д. Передача сообщений осуществляется через обменники, играющие роль асинхронных каналов связи. Они принимают, хранят и передают сообщения, соотносятся с входными и выходными портами задач. В состав ядра вводится диспетчер обменников, организующий прохождение сообщений через них.

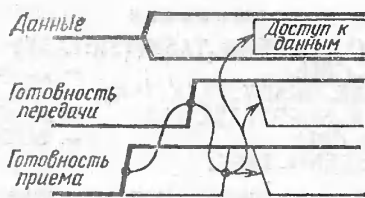


Рис. 3. Последовательность передачи данных

Различные типы обменников и стратегии их обслуживания моделируют те или иные физические каналы связи. Эти межзадачные интерфейсы отличаются числом задач, одновременно ожидающих и посылающих сообщения, типами сообщений и алгоритмами их распределения.

Чаще всего в микросистемах используются обменники, моделирующие каналы связи с памятью. Обмен сообщениями заключается в передаче доступа к блоку данных типа MESSAGE от одной задачи к другой. Обмен предусматривает этап синхронизации готовности передаваемых данных с готовностью их принять (рис. 3). Ссылка на сообщение передается в момент синхронизации.

Структура такого обменника может иметь вид:

```
type EXCHANGE is access
  record
    NAME:IDENTIFICATOR;    --имя обменни-
    --ка
    FIRST_TASK:TASK;       --указатели спи-
    --ска
    LAST_TASK:TASK;        --WALT_LIST
    FIRST_MSG:MESSAGE;     --указатели спи-
    --ска
    LAST_MSG:MESSAGE;      --MSG_LIST
    --указатель
    NEXT:EXCHANGE;        --EXCH_LIST
  end record;
```

Все обменники системы образуют список EXCH_LIST, организуемый ячейками SYS.FIRST_EXCH и SYS.LAST_EXCH, а также полем NEXT обменника. В качестве стратегии обслуживания обычно выбирают процедуру типа FIFO.

Когда задача обращается к обменнику, в котором нет сообщений, она вносится в список ожидания WAIT_LIST и переводится в состояние ожидания. Список организуется указателями FIRST_TASK, LAST_TASK дескриптора обменника и указателем NEXT дескриптора задачи. В определенный момент времени задача может ожидать только в одном обменнике.

Когда сообщение попадает в обменник, где уже ожидает несколько задач, оно передается первой из них в списке WAIT_LIST. Задача удаляется из списка и переводится в состояние готовности. Если ожидающая задача обладает более высоким приоритетом, чем передающая, то она становится активной. Если же при поступлении сообщения список WAIT_LIST пуст, то сообщение заносится в очередь MSG_LIST, указателями начала и конца которой служат поля FIRST_MSG и LAST_MSG дескриптора обменника. Ссылка на следующий элемент в очереди MSG_LIST содержится в поле NEXT сообщения.

При обращении задачи к обменнику с пустым списком MSG_LIST, ей передается указатель первой из них. Переданное сообщение из списка удаляется. В обменнике не могут одновременно находиться и задачи и сообщения. Приведенная на рис. 2 система содержит два обменника. Первый обменник имеет список ожидающих задач, а второй — список сообщений.

Возвращение доступа к области памяти, занятой сообщением, организуется посылкой обратного сообщения.

Обменник ответа обычно определяет посылающая задача, указывая его в поле RESPONSE сообщения. Обратным сообщением — подтверждением завершается полный цикл обмена между задачами. Для коллективно используемого ресурса MC создается одно представляющее его сообщение. Задача, владеющая сообщением, имеет право преимущественного доступа к MC-ресурсу. Освобождая ресурс она возвращает сообщение в обменник, в котором организуется очередь задач. Таким образом решается ключевая функция систем РВ — организация взаимоиключения.

Примитивы управления обменом. Базовый набор примитивов управления обменом содержит следующие элементы.

Примитив создания обменника

```
procedure RQ_CREATE_EXCHANGE(EXCH:EXCHANGE)
begin SAVE_CPU;
  ENTER_EXCH(EXCH);    --запись EXCH в
  --EXCH_LIST
  LOAD_CPU;
end RQ_CREATE_EXCHANGE;
```

Примитив уничтожения обменника

```
procedure RQ_DELETE_EXCHANGE(EXCH:EXCHANGE)
begin SAVE_CPU;
  DELETE_EXCH(EXCH);   --удаление EXCH из
  --EXCH_LIST
  LOAD_CPU;
end RQ_DELETE_EXCHANGE;
```

При уничтожении обменника ожидающим задачам посылается специальное системное сообщение, а очередь сообщений аннулируется. Возможен случай сбора неиспользованных сообщений в специальном обменнике.

Примитив посылки сообщения

```
procedure RQ_SEND_MESSAGE(EXCH:EXCHANGE,
  MSG:MESSAGE)
begin SAVE_CPU;
  if EXCH.FIRST_TASK = null
  then ENTER_MSG(MSG)
  else begin SEND(MSG,EXCH.FIRST_TASK);
    ENTER_READY_TASK(EXCH.FIRST_TASK);
    DELETE_WAIT_TASK(EXCH.FIRST_TASK);
  end;
  LOAD_CPU;
end RQ_SEND_MESSAGE;
```

Примитив получения сообщения

```
procedure RQ_RESIVE_MESSAGE(EXCH:EXCHANGE,
  MSG:MESSAGE)
begin SAVE_CPU;
  if EXCH.FIRST_MSG = null
  then begin ENTER_WAIT_TASK(EXCH, SYS.FIRST_READY);
    DELETE_READY_TASK(SYS.FIRST_READY);
  end
  else begin SEND(EXCH.FIRST_MSG,SYS.FIRST_READY);
    DELETE_MSG(EXCH.FIRST_MSG);
  end;
  LOAD_CPU;
end RQ_RESIVE_MESSAGE;
```

Для предотвращения зависания задач вводится модификация примитива получения сообщения, ограничивающая время возможного ожидания сообщения:

```
RQ_RESIVE_MESSAGE(EXCH:EXCHANGE, MSG:
  MESSAGE, TIME:INTEGER). Для контроля за временем ожидания задачи формируется дополнительный список задач DELAY_LIST, организуемый указателями
```


SYS.FIRST_DELAY, SYS.LAST_DELAY и полем NEXT_DELAY в дескрипторе задачи в порядке возрастания времени ожидания так, что первой всегда освобождается первая задача списка. Время ожидания TIME кодируется в поле DELAY дескрипторов задач из списка DELAY_LIST.

По окончании времени ожидания задаче возвращается код специального сообщения и она удаляется из списков DELAY_LIST и WAIT_LIST. Для определения списка ожидания в дескрипторе задач имеется поле EXCH_WAIT.

В некоторых исполнительных ядрах разрешается подвешивать любые задачи, включая ожидающие. В этом случае признак подвешивания SUSP заносится в поле STATUS дескриптора задачи, хотя сама задача остается в списке ожидания. При удалении из списка WAIT_LIST проверяется поле STATUS и, если был установлен признак подвешивания, задача заносится в список SUSP_LIST.

Обработка прерываний. Для обслуживания аппаратных прерываний в состав ядра вводится специальный циклический процесс — обработчик прерываний, функционирующий параллельно остальной части ядра. При этом необходимы механизмы взаимного исключения одновременного доступа к управляющим структурам системы со стороны этих двух составляющих ядра. Наиболее простой путь состоит в запрете прерываний при любом вхождении в ядро. Если время исполнения вызова велико, такая блокировка прерываний может оказаться неприемлемой. Возникает задача согласования работы обработчика прерываний с остальной частью ядра.

Основная идея организации работы обработчика прерываний состоит в том, чтобы генерацию аппаратного прерывания связать с посылкой некоторого сообщения — программного эквивалента сигнала прерывания. Для этого при входе в обработчик прерываний моделиру-

ется процедура сохранения текущего контекста SAVE_CPU, что обеспечивает возврат к прерванной задаче. Затем обработчик идентифицирует источник запроса и посылает соответствующее сообщение в заранее определенный обменник прерывания. Если у обменника есть ждущая задача, то она переводится в состояние готовности. Затем обработчик прерываний выполняет процедуру выхода из ядра LOAD_CPU, передавая управление наиболее приоритетной задаче. В качестве последней может быть задача, получившая сообщение о прерывании.

Каждому сигналу прерывания соответствует своя задача обслуживания прерывания и свой обменник, в который посылается сообщение о запросе. Задача обычно находится в состоянии ожидания возле обменника прерывания и переходит в состояние готовности после посылки соответствующего сообщения о прерывании. Приоритет задачи должен соответствовать приоритету запроса. В набор примитивов ядра обычно вводят ряд специальных запросов, позволяющих динамически управлять работой и составом обработчика прерываний, в сильной степени зависящих от используемой системы прерываний и характера решаемых задач.

ЛИТЕРАТУРА

1. Шоу А. Логическое проектирование операционных систем: Пер. с англ.— М.: Мир, 1981.—360 с.
2. Зелкович М., Шоу А., Гэннон Д.ж. Принципы разработки программного обеспечения: Пер. с англ.— М.: Мир, 1982, с. 197—250.
3. Квигглер П. Задачи, программы, вычисления, результаты: Пер. с англ.— М.: Мир, 1980, т. 280—357.
4. Электроника, 1982, № 8, с. 61—66.
5. Микропроцессорные средства и системы, 1984, № 2, с. 28—30.
6. Электроника, 1982, № 23, с. 103.

Статья поступила 29 марта 1985 г.

УДК 681.3.06.681.2

Д. А. Лукьянов

КАК НАПИСАТЬ КРОСС-ТРАНСЛЯТОР С ЯЗЫКА АССЕМБЛЕР

Вытесняя системы с «жесткой» логикой, построенные из простых логических элементов, микропроцессорные средства широко внедряются в современные электронные приборы. Сроки разработки и успешное функционирование создаваемого прибора со встроенным МП в основном определяются средствами подготовки и трансляции прикладного программного обеспечения (ППО) микропроцессорных средств.

В зависимости от сложности поставленной задачи объем и методы разработки ППО различаются.

Сложные задачи (десятки и сотни килобайт) обычно разрабатываются на языках высокого уровня с помощью специально созданных программных средств. Однако такие задачи встречаются, лишь если на базе мощного микропроцессора строится новая ЭВМ или вычислительная система.

Для массовых потребителей микропроцессоров программирование — один из этапов

разработки нового прибора со встроенным МП. Программирование на языке ассемблер позволяет наиболее полно учесть особенности схемной реализации. В этих случаях созданием ППО (10...15 Кбайт) нередко занимаются разработчики аппаратуры. Это позволяет получить наиболее эффективное сочетание программных и аппаратных компонент. Совмещение обязанностей особенно часто встречается в условиях мелкосерийного производства и разработки уникального оборудования. Описанные ниже программные средства предназначены прежде всего для таких условий.

Обычно программы, написанные разработчиком на языке ассемблера, разрабатываемая МП-система неспособна самостоятельно транслировать в машинные коды. Ручная трансляция непроизводительна. Отвергнув ее, приходится выбирать между двумя вариантами: либо получить в распоряжение микро-ЭВМ с той же системой команд, как и созда-

ваемая конструкция, либо воспользоваться уже имеющейся ЭВМ другого типа и поручить трансляцию и отладку кросс-системе.

Первый путь возможен для массовых микропроцессоров с жесткой системой команд (КР5801К80, МС68000, i8086, МОСТЕК 6502 и т. п.), но неприемлем для микропрограммируемых и малоразрядных МП.

Большую гибкость при меньших дополнительных затратах обеспечивает трансляция программ на уже имеющихся ЭВМ в кросс-режиме. В состав кросс-систем программирования обычно входят трансляторы с ряда языков, программы редактирования текста, математические модели-имитаторы разрабатываемых устройств, программы обратной трансляции (дисассемблеры) и другие сервисные программы.

Развитые кросс-системы для массовых микропроцессоров имеются на ЕС ЭВМ и БЭСМ-6. Использование кросс-систем на «больших» ЭВМ имеет существенный недостаток: полученный машинный код нельзя непосредственно загрузить в память прототипа прибора и, убедившись в неработоспособности программы, оперативно устранить найденные ошибки в ее исходном тексте.

Удобнее воспользоваться дешевыми мини- и персональными ЭВМ, позволяющими наряду с трансляцией ППО загружать и отлаживать его на прототипе [1]. К сожалению, для ряда микропроцессоров (особенно специализированных) найти полностью удовлетворяющую разработчика систему не удается — в этом случае проще приспособить имеющиеся на ряде мини- и микроЭВМ программы из системного ПО для работы в кросс-режиме.

Рассмотрим кросс-систему (рис. 1) на базе широко распространенных в лабораторной практике ЭВМ с системой команд PDP-11 (СМ-3, СМ-4, «Электроника 60», ДВК) и операционной системы (ОС) РАФОС [2]. В составе этой ОС имеются развитые средства работы с файлами, редактирования текста, создания библиотек и документирования программ, в готовом виде пригодные для кросс-системы.

Макроассемблер МАКРО-11, входящий в РАФОС, можно приспособить для работы в качестве кросс-транслятора ППО с языка ассемблера (КТА) для применяемого типа МП БЕЗ ИЗМЕНЕНИЙ в самом макроассемблере (рис. 1). Получившийся КТА вполне пригоден для программирования большинства микропроцессоров с разрядностью адреса и машинного слова до 16. Остальные компоненты кросс-системы (рис. 2) написаны на языке Фортран и ассемблер, и поэтому здесь не рассматриваются.

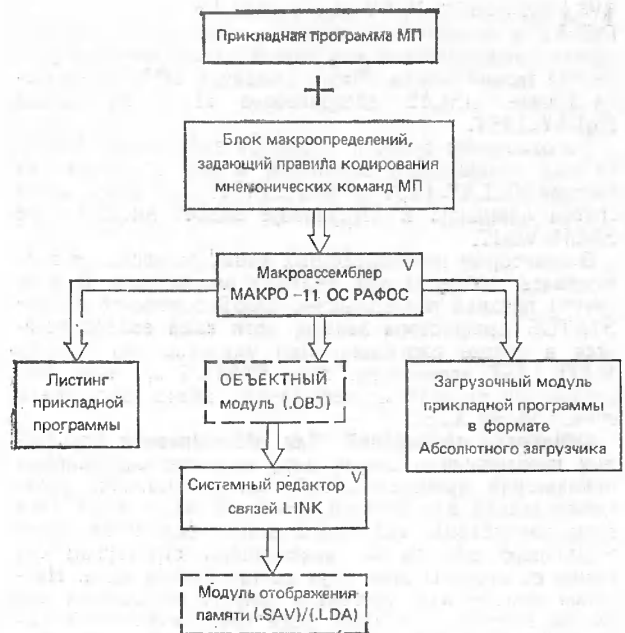


Рис. 1. Схема использования микроассемблера в кросс-режиме: V — имеется в ОС РАФОС

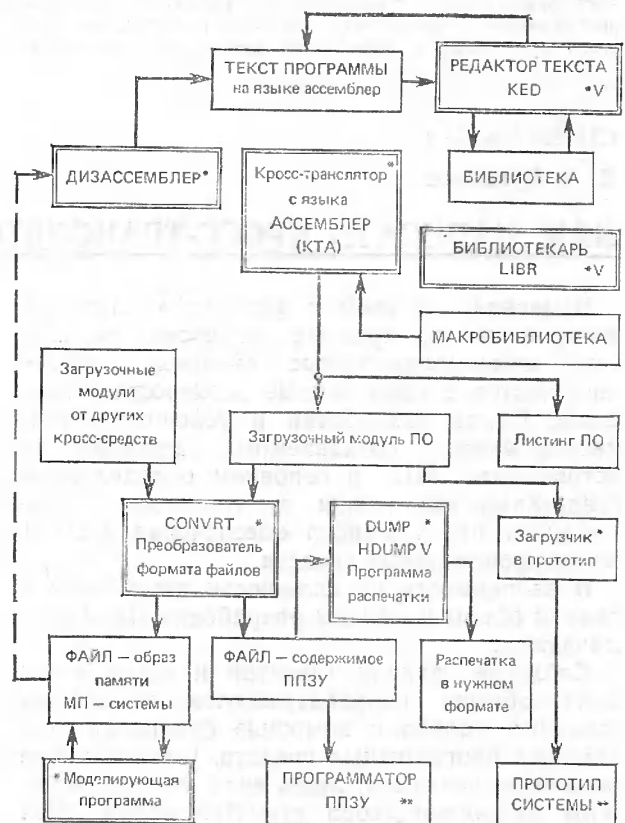


Рис. 2. Структура кросс-системы разработки программного обеспечения: * — программные компоненты, ** — аппаратные компоненты, V — имеется в ОС РАФОС

КТА как блок макроопределений

Несмотря на разнообразие систем команд существующих МП, основная функция транслятора с языка ассемблер (ТА) для любого из них — однозначное преобразование мнемонических выражений в машинные коды и правильное размещение их в памяти (рис. 3), причем каждой команде может соответствовать в общем случае разное число слов машинного кода. ТА, пользуясь счетчиком слов и таблицами имен, заменяет текст программы последовательностью двоичных кодов.

ТА применяемой микроЭВМ можно превратить в кросс-ассемблер, задав правила генерации кодов через «нестандартные» для этого транслятора мнемонические выражения и уже имеющиеся внутренние счетчики и таблицы. Для этого подходят ТА с широким набором макросредств и встроенных математических операций и разрядностью внутренних счетчиков адреса не менее разрядности адреса МП, для которого создается КТА. Чтобы изменить правила генерации кодов в *МАКРОАССЕМБЛЕРАХ*, не нужно изменять саму программу ТА. Достаточно присвоить мнемонические имена команд микропроцессора макроопределениям. Тогда каждый раз, когда в тексте прикладной программы встретится мнемокоманда МП, вместо ее непосредственной трансляции будет вызвано соответствующее макроопределение. В результате в транслируемую программу вводится текст макроопределения, в котором формальные параметры заменены их значениями, определенными или вычисленными во время первого просмотра текста ТА.

Если систему макроопределений написать так, чтобы они сводили текст прикладной программы МП только к машинно-независимым операциям записи чисел по заданному или текущему адресу трансляции (а такая возможность имеется во всех макроассемблерах для микроЭВМ), то независимо от системы команд, на которую рассчитан исходный ТА, он будет работать в кросс-режиме.

Трансляция команд теперь распадается на две фазы. В первой фазе макроассемблер играет роль диспетчера макровывозов, «передавая управление» макроопределениям с соответствующим именем. Во второй фазе выбранное МО генерирует машинный код МП, в зависимости от типа и значений аргументов. Макроопределения здесь эквивалентны программам-интерпретаторам отдельных мнемокоманд.

Макроопределения выполняют еще несколько важных функций кросс-ассемблера: анализ правильности мнемоники команд и синтаксиса предложений исходного текста,



Рис. 3. Схема работы транслятора с языка ассемблер

диагностику специфических ошибок, предусмотренных автором, и вывод соответствующих сообщений в листинг прикладной программы или на терминал оператора. Для этого в исходном ТА необходима возможность условной трансляции.

Следовательно, именно в макроопределениях нужно отразить особенности конкретного МП; принятую мнемонику команд и имена регистров, порядок аргументов и их допустимые значения, и т. п. Весь КТА сводится к индивидуальному для каждого МП блоку макроопределений, вместе с которым транслируется текст программы с мнемоникой команд этого МП.

При таком подходе к построению КТА не придется самостоятельно программировать механизмы меток и внутренних таблиц трансляции — наиболее сложную часть кросс-ассемблеров, написанных на других языках. К тому же мы сразу получим кросс-макроассемблер, поскольку возможностями макроопределений можно воспользоваться и в тексте прикладной программы. Так как макроопределения пользователя будут содержать мнемокоманды МП, допустимая глубина вложения макровывозов при этом уменьшится на число вложений, использованных при определении команд МП.

Адаптация МАКРО-11

Применим этот подход к макроассемблеру МАКРО-11 [2] операционных систем РАФОС и ОС РВ, предположительно хорошо знакомому читателю. Напомним, что МАКРО-11 имеет 16-разрядные счетчики адреса трансляции, и:

обеспечивает работу с константами в двоичном, восьмеричном и десятичном представлении;

автоматически переводит текстовые строки и отдельные символы в коды КОИ-7;

позволяет сравнивать мнемонические выражения по тексту и по значению, а также обеспечивает вычисление арифметических и логических выражений;

имеет возможность построения блоков определенных и неопределенных повторов, блоков с локальными метками, и автоматически генерирует локальные метки в макроопределениях;

имеет большой набор директив управления трансляцией и листингом ее результатов, размещаемых в исходном тексте;

позволяет одновременно вести до 256 программных секций с независимыми счетчиками адреса трансляции в каждой из них.

Объем внутренних таблиц МАКРО-11 не ограничен доступной оперативной памятью, и если ее объем недостаточен, таблицы автоматически продолжают в внешних временных файлах.

В МАКРО-11 имеются стандартные операции и директивы, полезные для «описания» команд микропроцессоров:

*2^N — логический сдвиг значения мнемонического выражения влево на N разрядов;

1/2^N — то же, но вправо;

&MASK — логическое умножение значения операнда на маску, чтобы выделить отдельные биты;

AIB — логическое сложение, полезное для окончательной сборки машинного кода;

.IF/.ENDC — директивы условной трансляции;

.BYTE X — занесение байта X в объективный модуль по текущему адресу. Значение счетчика адреса трансляции увеличивается на 1;

.WORD X — занесение слова X в объективный модуль по текущему адресу. Значение счетчика адреса трансляции увеличивается на 2.

Пользуясь этими операциями, нетрудно написать первые макроопределения команд, например для 8-разрядной однокристальной микроЭВМ К1816ВМ48. Команда безусловного перехода JMP (рис. 4) кодируется двумя последовательными байтами, причем первый содержит код операции и три старших разряда адреса перехода, а второй — восемь младших разрядов этого адреса. Правила кодирования записаны в макроопределении .MACRO JMP, поэтому теперь (при появлении в тексте прикладной программы JMP LABEL) МАКРО-11, пользуясь найденным значением адреса метки LABEL, внесет два байта машинного кода, начиная с текущего адреса трансляции. Чтобы текст макроопределений не попадал в листинг прикладной программы, они обрамляются директивами .NLIST SRC и .LIST SRC соответственно.

Команда пересылки MOV микроЭВМ К1816ВМ48 имеет девять различных вариантов. Чтобы МАКРО-11 автоматически выбирал нужный, воспользуемся условной трансляцией, т. е. будем транслировать разные фрагменты макроопределения в зависимости от условий его вызова. Следующее макроопределение (рис. 5) анализирует способы адресации аргументов вспомогательной макро-

```

.NLIST
* ПИСОМАНН ТЕРСТ
* ФОРМАТ КОМАНДЫ JMP ADDR ОДНОКРИСТАЛЬНОЙ
* МИКРОЭВМ КР1816ВМ48
*
* БАЙТ 1 !A10!A9!A8! 0! 0! 1! 0! 0!
*
* БАЙТ 2 ! A? !A6!A5!A4!A3!A2!A1!A0!
*
* "ОПИСАНИЕ" КОМАНДЫ JMP
.MACRO JMP ADDR
.NLIST SRC
.BYTE <4!<<ADDR/10>>&340>>>><ADDR&377>
.LIST SRC
.ENDM JMP
.LIST
.LIST
* ПРИКЛАДНАЯ ПРОГРАММА
*
* .ASECT
* .=1000 ; ЗАДАНИЕ АДРЕСА
* ; ТРАНСЛЯЦИЯ
BEGIN: JMP OUT
OUT: JMP BEGIN
.END
*
* СБРАСЕТ ЛИСТИНГА ПРОГРАММЫ
*
* ПРИКЛАДНАЯ ПРОГРАММА
000000 .ASECT
001000 .=1000
001000 BEGIN: JMP OUT
001000 104 002
001000 OUT: JMP BEGIN
001002 104 00B
001002 000001 .END

```

Рис. 4. «Описание» команды и пример листинга программы для кросс-ассемблера для 8-разрядной однокристальной микроЭВМ К1816ВМ48

командой .MODE, сравнивающей фактический текст аргумента X с возможными символами типа адресации, и в результате присваивает временной переменной MODE значения 0..2. В зависимости от значения этой переменной .MACRO MOV генерирует различные варианты машинного кода. Для упрощения анализа аргументов знак способа адресации в тексте прикладной программы отделяется от операнда пробелом, и МАКРО-11 воспринимает их как два независимых аргумента. Поэтому макроопределение .MOV имеет не два, а четыре формальных параметра.

Для интерпретации варианта команды рассматриваемое макроопределение анализирует

```

.MACRO .MODE
* ОПРЕДЕЛЕНИЕ СПОСОБА
* АДРЕСАЦИИ АРГУМЕНТА
* .IF IDN %3
MODE =2
* .IFF
* .IF IDN %2
MODE =1
* .IFF
MODE =0
.ENDC
.ENDM
*
* .MACRO MOV A,B,C,T
* ПЕРЕБОР ВАРИАНТОВ
* КОМАНДЫ MOV
* .MODE A
* .IF EQ MODEB
* .IF IDN A,PSIA
* =====> MOV PSIA B
* .BYTE 327
* .NEXT
* .MODE B
* .IFF
* .IF NE MODEC
* =====> MOV A, B AND
* .BYTE 300!C&377
* .NEXT

```

Рис. 5. Фрагмент макроопределения команды пересылки для 8-разрядной однокристальной микроЭВМ К1816ВМ48

аргументы не самым изящным методом линейного перебора, а когда необходимый вариант найден, обработка макроопределения прерывается директивой .MEXIT (аналог конструкции IF/THEN Паскаля). Можно построить макроопределение и так, чтобы сначала все возможные признаки собирались в одну переменную, а затем нужный вариант выбирался табличным методом (аналогично конструкции CASE Паскаля). Если необходимый вариант команды найден, макроопределения генерируют последовательность машиннонезависимых транслируемых директив .BYTE, .WORD, .ASCII, .ASCIZ, которые заносят машинный код микропроцессора в грузочный модуль.

Если разрядность микропроцессора не кратна восьми битам, для хранения каждого машинного слова МП придется резервировать несколько байт, а правильные «физические» адреса меток получатся после коррекции значения их адреса, взятого из внутренних таблиц МАКРО-11. Именно так написано макроопределение для команды JMS 12-разрядного процессора с системой команд мини-ЭВМ «Электроника 100И», где аргумент ADDR сдвигается на один бит влево. Другая особенность макроопределения MACRO JMS (рис. 6) состоит в том, что при прямой адресации JMS ADDR фактическое значение адреса перехода будет присвоено не формальному аргументу ADDR, а MODE. Анализируя аргументы команды, макроопределение генерирует код либо для прямой, либо для косвенной адресации.

```

:          ФОРМАТ КОМАНДЫ JMS ЭВМ "ЭЛЕКТРОНИКА-100И"
:
:  _____
:  |НЕ ИСП. |1|1|0|1|0|1|БИТ|СТР| АДРЕС НА СТРАНИЦЕ |
:  |-----|-----|-----|-----|-----|-----|
:  |15      |11|10| 9| 8| 7| 6|                                     |0
:
: MACRO JMS      MODE, ADDR
: .LIST SRC, CMD
: .IF     JMN    MODE, 1 ;==> КОСВЕННАЯ АДРЕСАЦИЯ
: .IF     EQ    <ADDR/2>&7600 ; НУЛЕВАЯ СТРАНИЦА
: .WORD  4000!400!<<ADDR/2>&177>
: .IF     NE    <ADDR/2>&177 ; ТЕКУЩАЯ СТРАНИЦА
: .WORD  4000!600!<<ADDR/2>&177>
: .ENDC
: .IFF
: .IF     EQ    <MODE/2>&7600 ; ПРЯМАЯ АДРЕСАЦИЯ
: .WORD  4000!<<MODE/2>&177> ; НУЛЕВАЯ СТРАНИЦА
: .IF     NE    <MODE/2>&177 ; ТЕКУЩАЯ СТРАНИЦА
: .WORD  4000!200!<<MODE/2>&177>
: .ENDC
: .LIST SRC, CMD
: .ENDM JMS
:
: .LIST MSG

```

Рис. 6. Фрагмент кросс-ассемблера для 12-разрядного процессора с системой команд мини-ЭВМ «Электроника 100И»

Диагностика ошибок

Полезность автоматической трансляции увеличится, если ТА сможет отыскивать логические и синтаксические ошибки в программе. Большинство грубых ошибок найдет

МАКРО-11: неопределенные или многократно определенные метки, неверная мнемоника или недопустимые аргументы вызовут ошибки макрорасширения и будут отмечены флагами ошибок в листинге [3]. Но необходимо учесть и ряд специфических ошибок, которые МАКРО-11 не обнаруживает самостоятельно, и заложить соответствующий анализ в макроопределения. Для каждого конкретного МП возможные ошибки будут различными, причем соответствующие макроопределения могут не только обнаруживать, но и в необходимых случаях по желанию программиста исправлять найденные ошибки. Так, в однокристалльной микроЭВМ К1816ВМ48 используется страничная адресация, поэтому ТА должен пресекать попытки программиста устраивать ветвления программы за пределы текущей страницы памяти командами условного перехода. В МП-системах с небольшим объемом ЗУ полезна диагностика недопустимых обращений за пределы отведенного адресного пространства и т. п.

Для диагностики ошибок в макроопределениях применяется условная конструкция вида:

```

: .IF <условие> <выражение>
: .PRINT <текст диагностического сообщения>
: .ENDC
или
: .IF <условие> <выражение>
: .ERROR <текст диагностического сообщения>
: .ENDC

```

Так можно анализировать текстовые и числовые выражения, проверять допустимость имен регистров и т. п. Строки с директивами .PRINT и .ERROR всегда печатаются в листинге ППО. Директива .ERROR увеличивает на единицу содержимое счетчика ошибок МАКРО-11.

Кросс-макросассемблер для микропроцессора КР580ИК80

В полном тексте КТА МИКРОСС (рис. 7) описанный способ диагностики применен для анализа и исправления имен регистров, формата аргументов и недопустимых обращений к адресам вне области, ограниченной LOADDR и HIADDR соответственно.

Особенность КТА состоит в том, что макроопределения большинства команд не выписаны явно, а тоже генерируются при вызове кросс-ассемблера по правилам, заданным макроопределениями ONEBYT, TWOBYT и THREEBYT, в результате чего готовый текст КТА «разворачивается» только на этапе трансляции. Такое сокращение текста КТА возможно не для любого МП.

```

ENABL LC
*SETTL MICROSS - КРОСС-АСЕМБЛЕР КР5801К80А U2.04А
*MLIST
*ENASLE ABS ;<- ВВЕСТИ ПРН АБСОЛЮТНОЙ ТРАНСЛЯЦИИ
*MLIST MD,TOC,CND
*НИЖНЯЯ ГРАНИЦА ДОПУСТИМЫХ АДРЕСОВ ДЛЯ ДИАГНОСТИКИ
*IF NE MD,LOADR, LOADDR=0 ; OSTAT
* ВЕРХНЯЯ ГРАНИЦА ДОПУСТИМЫХ АДРЕСОВ ДЛЯ ДИАГНОСТИКИ
*IF NE HIADDR, HIADDR=17777; OSTAT
* ВХОД ЗА УСТАНОВЛЕННЫЕ ГРАНИЦЫ ВЕДЕТ К ВЫВОДУ
* СООБЩЕНИЯ ОБ ОШИБКЕ КРОСС-ТРАНСЛЯТОРИ.
*ISABL REG
E=0 ; ВХОДИТ ИМЕНОНУЮ РЕГИСТРОВ МИКРОПРОЦЕССОРА
E=1 ; КР5801К80А, ВСЕ ОБОЗНАЧЕНИЯ РЕГИСТРОВ
E=2 ; ИАСРО-11 ТЕПЕРЬ НЕ ДЕЙСТВУЮТ.
E=3 ; НЕОБХОДИМО, ЧТОБЫ ИМЕНА, ВХОДЯЩИЕ
E=4 ; В ПРОГРАММУ, НЕ СООБЛАДИ С ИМЕНАМИ
E=5 ; РЕГИСТРОВ.
E=6 ;
E=7 ;
E=PM=6 ;
E=SP=6 ;

```

ОПИСАНИЕ ОСНОВНЫХ КОМАНД МИКРОПРОЦЕССОРА

```

*MACRO RST * ; ПРОГРАММНЫЕ ПРЕРЫВАНИЯ
*MLIST SRC
*IF NE X<X&70>
*ERROR X ;*** НЕВЕРНАЯ ААРЕСА RST
*ENDC
*BYTE <3B7+X&70>
*LIST SRC
*ENDM RST

*MACRO MOV TO,FROM ; ПЕРЕХОДИТ
*MLIST SRC
*IF NE <TO-<TO&7>><FROM-<FROM&7>>
*ERROR TO,FROM ;*** ОШИБКА ИМЕНИ РЕГИСТРА MOV
*ENDC
*BYTE <100+<TO&7>*10+<FROM&7>>
*LIST SRC
*ENDM MOV

*MACRO MVI TO,WHAT ; ЗАГРУЗКА РЕГИСТРОВ
*MLIST SRC
*IF NE TO-<TO&7>
*ERROR TO ;*** ОШИБКА ИМЕНИ РЕГИСТРА MVI
*ENDC
*BYTE <6+<TO&7>*10>><WHAT&377>
*IF NE WHAT-<WHAT&377>
*ERROR WHAT ;*** АРИФМЕТИЧЕСКИЙ АРГУМЕНТ MVI
*ENDC
*LIST SRC
*ENDM MVI

*MACRO LXI TO,WHAT ; ЗАГРУЗКА ПАРУ РЕГИСТРОВ
*MLIST SRC
*IF NE <TO-<TO&7>><TO&81>
*PRINT TO ;*** ОШИБКА ИМЕНИ РЕГИСТРА LXI
*ENDC
*BYTE <<<TO&7>/2>*20+1>><WHAT&377>><WHAT&177400+400>
*LIST SRC
*ENDM LXI

*MACRO ONEBYT INST,A,B,C ; АРИФМЕТИЧЕСКОЕ КОМАНДИ
*MACRO INST X ; КОМАНДИ
*MLIST SRC
*IF NE X<X&67>
*PRINT X ;*** ОШИБКА ИМЕНИ РЕГИСТРА INST
*ENDC
*BYTE <A+<B*<C&67>/C>>>
*LIST SRC
*ENDM INST
*ENDM ONEBYT

*MACRO TWOBYT INST,A ; АРИФМЕТИЧЕСКОЕ КОМАНДИ
*MACRO INST * ; КОМАНДИ
*MLIST SRC
*IF B= X
*BYTE C
*ENDC
*IF NE, Q
*IF EQ <C&200>
*BYTE C
*PRINT X ;*** АРИФМЕТИЧЕСКИЙ АРГУМЕНТ КОМАНДИ INST
*IF
*IF NE X<X&677>
*PRINT X ;*** АРГУМЕНТ АРИФМЕТИЧЕСКОЕ КОМАНДИ INST
*ENDC
*BYTE <C+X&677>
*ENDC
*ENDC
*LIST SRC
*ENDM INST
*ENDM TWOBYT

*MACRO THREEBYT INST,A ; ТРЕХАРИФМЕТИЧЕСКОЕ КОМАНДИ
*MACRO INST X
*MLIST SRC
*IF LT <HIADDR><HIADDR><HIADDR><X>>
*IF GT <X&77777><HIADDR&77777>
*ERROR X ;*** НИЖЕ ЗОНА ААРЕСАЦИИ
*ENDC
*ENDC

```

```

*IF LT <HIADDR>
*ERROR X ;*** НИЖЕ ЗОНА ААРЕСАЦИИ
*ENDC
*IF LT <LOADDR><LOADDR><X>>
*IF LT <X&77777><LOADDR&77777>
*ERROR X ;*** НИЖЕ ЗОНА ААРЕСАЦИИ
*ENDC
*IF LT <LOADDR><X>
*ERROR X ;*** НИЖЕ ЗОНА ААРЕСАЦИИ
*ENDC
*BYTE A,<X&377>><X&177400+400>
*LIST SRC
*ENDM INST
*ENDM THREEBYT

*КОМПЛИМЕНТАРНЫЕ КОМАНДИ
*MACRO CLC ; ОЧИСТКА БИТА ПЕРЕНОСА
*ORA A
*ENDM CLC

*MACRO LOAD ADDR,STACK SP ИЛИ HL
*IF
*LXI ADDR,STACK
*IF
*LXI ADDR,STACK
*ENDC
*ENDM LOAD

*MACRO CLA ; ОЧИСТКА АЛЮМИНАТОРА
*XRA A
*ENDM CLA

*MACRO BYTE *
*MLIST SRC
*BYTE X
*LIST SRC
*ENDM BYTE

*ЭТИ МАКРООПРЕДЕЛЕНИЯ - ДЛЯ СОВМЕСТИМОСТИ
*С ДРУГИМИ ТРАНСЛЯТОРАМИ.
*MACRO IS X
*BLKB X
*ENDM IS
*MACRO DB *
*BYTE *
*ENDM DB
*MACRO DC *
*ASCII *
*ENDM DC

```

```

ONEBYT STA,2,20,2 TWOBYT XTL,343
ONEBYT INR,3,20,2 TWOBYT ANI,346
ONEBYT INR,4,10,1 TWOBYT RPE,350
ONEBYT DCR,5,10,1 TWOBYT PHL,351
ONEBYT DAB,11,20,2 TWOBYT MCH,353
ONEBYT LDAB,12,20,2 TWOBYT MFI,356
ONEBYT DCR,13,20,2 TWOBYT RFI,360
ONEBYT ADD,200,1,1 TWOBYT DI,363
ONEBYT ADC,210,1,1 TWOBYT ORI,366
ONEBYT SUB,220,1,1 TWOBYT RMI,370
ONEBYT SBB,230,1,1 TWOBYT SPHL,371
ONEBYT ANI,240,1,1 TWOBYT EI,373
ONEBYT XRA,250,1,1 TWOBYT CPI,376
ONEBYT ORA,260,1,1
ONEBYT ORI,270,1,1
ONEBYT POP,301,20,2 THREEBYT SHLD,42
ONEBYT PUSH,302,20,2 THREEBYT LHL,52
ONEBYT STA,303,20,2 THREEBYT STA,62
THREEBYT LDA,72
TWOBYT NOP,0
TWOBYT RLC,7
TWOBYT RRC,17
TWOBYT RAL,27
TWOBYT RAR,37
TWOBYT DAB,47
TWOBYT CMA,57
TWOBYT STC,67
TWOBYT CMC,77
TWOBYT HL,160
TWOBYT RND,300
TWOBYT ADI,306
TWOBYT RZ,310
TWOBYT RET,311
TWOBYT ACI,316
TWOBYT RMI,320
TWOBYT OUT,323
TWOBYT SUI,326
TWOBYT RC,330
TWOBYT IN,333
TWOBYT SBI,336
TWOBYT RPI,340
TWOBYT XTL,343
TWOBYT ANI,346
TWOBYT RPE,350
TWOBYT PHL,351
TWOBYT MCH,353
TWOBYT MFI,356
TWOBYT RFI,360
TWOBYT DI,363
TWOBYT ORI,366
TWOBYT RMI,370
TWOBYT SPHL,371
TWOBYT EI,373
TWOBYT CPI,376
THREEBYT SHLD,42
THREEBYT LHL,52
THREEBYT STA,62
THREEBYT LDA,72
THREEBYT JNZ,362
THREEBYT JNP,363
THREEBYT CHD,364
THREEBYT JZ,312
THREEBYT CZ,314
THREEBYT CALL,315
THREEBYT JNC,322
THREEBYT CHC,324
THREEBYT JC,332
THREEBYT CC,334
THREEBYT JFO,342
THREEBYT CPO,344
THREEBYT JPE,352
THREEBYT CPE,354
THREEBYT JP,362
THREEBYT CP,364
THREEBYT JH,372
THREEBYT CH,374
***** C
*LIST NEW:BD
*LIST

```

Рис. 7. Полный текст кросс-асемблера для микропроцессора КР5801К80А

В прикладных программах можно использовать и ряд дополнительных мнемонических команд, отсутствующих в стандартном ассемблере БИС КР580ИК80, но часто встречающихся в программах и облегчающих ее чтение. Это — очистка аккумулятора (CLA), очистка регистра «С» (CLL), загрузка регистров HL и SP (LOAD). При последнем макровызове происходит контекстный анализ параметра ADDR, и если его имя совпадает с STACK, то этим значением загружается SP, а если не совпадает — то HL. Обе операции расширяются командами LXI. Часть макроопределений DB, DS, DW дублирует соответствующие директивы МАКРО-11 и введены для совместимости с мнемоникой для КР580ИК80.

Макробιβлиотеки

РАФОС имеет средства работы с библиотеками. Это позволяет компактно хранить часто используемые и отлаженные подпрограммы и их фрагменты. Системная программа LIBR включает и удаляет библиотечные модули, а директива .MCALL МАКРО-11 вызывает их в прикладных программах.

В макробιβлиотеках можно хранить и набор кросс-ассемблеров, если весь текст каждого КТА оформить как одно макроопределение и присвоить ему соответствующее имя, например .CROSS. Тогда для вызова КТА в начале прикладной программы достаточно поместить директивы .MCALL CROSS и .CROSS.

Оформление и трансляция прикладной программы

Поскольку для трансляции служит МАКРО-11, текст программы оформляется (как любая программа на ассемблере СМ ЭВМ) с использованием его стандартных директив, но мнемоника команд соответствует применяемому типу МП. Полученный текст не полностью удовлетворяет всем соглашениям ассемблеров (если таковые имеются) конкретных МП в отношении обозначений меток и операторов присваивания. Здесь сохраняются все правила МАКРО-11, поэтому необходимые коррекции очевидны для программистов СМ ЭВМ.

Директива .ASECT в начале текста ПП позволяет распределять память вручную и сразу получать загрузочный модуль, если вся прикладная программа размещена в абсолютной секции. Файл кросс-ассемблера и файл программы (порядок важен!) объединяются и обрабатываются макроассемблером (см. рис. 1). В результате получается листинг (см. рис. 4) с указанными в нем физическими адресами и машинным кодом, и загрузочный модуль программы в формате Аб-

солютного загрузчика, который можно либо передать в память МП, либо с помощью программы-имитатора загрузчиков CONVRT преобразовать в двончный файл-образ содержимого памяти и сохранить для дальнейшей отладки или записи в ППЗУ. Этот вариант трансляции предпочтителен для создания простых программ при небольшом опыте программирования.

Заметим, что МАКРО-11 совместно с системным компоновщиком РАФОС LINK позволяет создавать модульные программы для МП, окончательно связываемые и размещаемые в памяти только на этапе компоновки точно так же, как и программы для применяемой ЭВМ. При этом в исходном тексте может быть одна абсолютная и до 255 именovaných программных секций, в каждой из которых допустимы любые (возможно, совпадающие) локальные имена и метки, а связь между секциями осуществляется заданием глобальных имен.

В кросс-режиме это возможно потому, что макроассемблер производит только предварительную обработку мнемонических выражений, если в них встречаются неопределенные глобальные имена, и включает в объектный модуль (см. рис. 1) информацию о правилах вычисления кодов через определенные компоновщиком фактические значения глобальных имен. Проверки на допустимость обращения (см. рис. 6) теперь вызовут ошибки, так как адреса на этапе трансляции определены только в абсолютной секции. Этим вариантом трансляции, проверенным для КТА МИКРОСС, можно пользоваться, лишь получив полное представление о работе компоновщика РАФОС. LINK не вводит информацию для монитора РАФОС в загрузочный модуль, если использовать ключ /LDA.

Рассмотренный способ построения кросс-трансляторов с языка ассемблер в виде блока макроопределений прост и экономичен, а сохранение директив и синтаксиса исходного транслятора МАКРО-11 создает дополнительные удобства для разработчика микропроцессорной системы, знакомого с программированием микроЭВМ. Предложенный подход применим и для других макроассемблеров (например, M80 операционной системы CP/M80) и не исключает, а дополняет другие способы построения КТА.

ЛИТЕРАТУРА

1. Иванов Е. А., Муренко Л. А., Широков Ю. Ф. Универсальная отладочная система автоматизации проектирования микропроцессорных устройств. — Микропроцессорные средства и системы, 1984, № 3, с. 53.

2. Снйгер М. Мини-ЭВМ PDP-11: Программирование на языке ассемблера и организация машины: Пер. с англ. — М.: Мир, 1984. — 272 с., ил.

Статья поступила 15 марта 1985 г.

Ускоренно развивать выпуск средств автоматизации инженерного труда, малых электронно-вычислительных машин высокой производительности, персональных электронно-вычислительных машин, систем числового программного управления для многофункциональных станков и гибких производственных модулей, программируемых командо-аппаратов для различных видов оборудования.

Из проекта Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года

УДК 681.3.06:322.1

В. В. Липаев, Ф. А. Каганов, А. В. Керданов, Ю. В. Загубин, А. Э. Колдобский, А. С. Новичков

СИСТЕМА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ПРОГРАММ НА БАЗЕ ПЕРСОНАЛЬНЫХ ЭВМ (СИСТЕМА ПРА)

Введение. Применение персональных микроЭВМ (ПЭВМ) в различных отраслях народного хозяйства обуславливает широкий спектр технологий разработки программ и разнообразие технологического программного обеспечения. Для разработки небольших вычислительных или сервисных программ применяются языки программирования и операционные системы, пока не объединенные строго регламентированной технологией проектирования программ. В некоторых случаях вводятся ограничения на используемые языки программирования и методические инструкции при создании комплексов программ средней сложности. Однако этого недостаточно для разработки сложных комплексов программ, как продукции производственно-технического назначения с определенным качеством и документацией. Необходимы профессиональные технологические системы, обеспечивающие промышленную разработку программ для различных типов микроЭВМ. При этом микроЭВМ могут использоваться в качестве технологических для разработки программ и в качестве ЭВМ, реализующих эти программы по их функциональному назначению.

Для микроЭВМ, встраиваемых в объект управления, ресурсы которых ориентированы на решение задач конкретного применения, часто необходимы автономные по отношению к объекту средства автоматизации проектирования программ — технологические кросс-системы, базирующиеся на ЭВМ с большими ресурсами и различными системами команд, в комплексе с ПЭВМ. Создание локальной сети на базе ПЭВМ и универсальной машины и распределение между ними функций автоматизации проектирования позволяет значительно повысить общесистемную надежность и улучшить эксплуатационные характеристики технологического комплекса. Одновременно значительно увеличиваются вычислительные ресурсы, прежде всего доступные объемы оперативной и внешней памяти. Кроме того, снижается интенсивность обмена с внешней памятью в универ-

сальной ЭВМ, что повышает эффективность использования процессоров и сокращает время отклика на задания пользователей.

Использование профессиональных персональных ЭВМ для автоматизации проектирования программ. Кросс-системы на базе автономных ПЭВМ, ориентированные на проектирование комплексов программ малой и средней сложности. Ограниченные ресурсы существующих ПЭВМ позволяют иметь только адаптируемые кросс-трансляторы с ассемблера и макроязыка, а также средства тестирования модулей и небольших групп программ в режиме интерпретации. Состав документации ограничен минимумом эксплуатационных документов на программы. Изготовление разнообразных машинных носителей трудно реализовать, поэтому для передачи готовых программ используется перфолента или электрический канал. В целом объем программ кросс-систем пока ограничен размерами внешней и оперативной памяти ПЭВМ.

Кросс-системы на базе ПЭВМ, комплексированных в локальную сеть с большими и универсальными ЭВМ, используют преимущества технологических ЭВМ обоих классов и позволяют создавать комплексы программ высокой сложности. В этом случае наиболее простые функции автоматизации разработки небольших программных компонент возлагаются на средства ПЭВМ, а сложные функции комплексирования модулей и отладки больших комплексов программ переносятся на универсальную ЭВМ. Распределение функций между ними зависит от создаваемого комплекса программ и определяет варианты взаимодействия комплексированных ЭВМ в локальной сети. В минимальном варианте универсальная ЭВМ может использоваться для автоматизированной адаптации кросс-системы, а также для выпуска некоторых видов документов и машинных носителей. В варианте полного функционального комплексирования универ-

сальная ЭВМ используется для сопряжения программных модулей и групп программ, для их комплексной отладки, выпуска всех документов и машинных носителей, а на ПЭВМ отрабатываются модули и небольшие группы программ, полная база данных проектирования которых размещается на универсальной ЭВМ.

Персональные ЭВМ могут также использоваться в качестве стенда для имитации внешней информации (например, роботов или ГАП) и обработки результатов функционирования комплексов программ. Для этого ПЭВМ сопрягаются каналами связи с микроЭВМ конкретных систем управления и обработки информации. Комплексная отладка и испытания программ в этом случае производятся на реальной микроЭВМ в реальном или псевдореальном масштабах времени.

Таким образом, на существующих ПЭВМ могут быть созданы унифицированные системы автоматизации проектирования, испытаний и сопровождения комплексов программ малой и средней сложности для различных типов микроЭВМ, используемых для управления и обработки информации. Для разработки таких систем эффективно использовать принципы ПРОМЕТЕЙ-технологии [1].

Основные концепции ПРОМЕТЕЙ-технологии. ПРОМЕТЕЙ-технология и ее инструментальные системы предназначены для разработки программной продукции производственно-технического назначения, базирующейся на встраиваемых специализированных мини- и микроЭВМ, используемых для автоматизации технологических процессов, роботов, гибких автоматизированных производств, энергетических комплексов и т. д. Особенности этого класса программ состоят в следующем:

исполняются в реальном масштабе времени;

обладают высокой «чувствительностью» к поступающим внешним воздействиям;

эффективно используют ресурсы ЭВМ для решения функциональных задач;

размещаются и функционируют на ЭВМ с различными характеристиками (память, производительность, система команд);

структурированы, т. е. состоят из сотен модулей, имеющих глубокую связность по информации и управлению, требуют специальных работ по комплексированию;

имеют необходимые фиксированные показатели качества и надежности функционирования, достаточные для применения по назначению;

должны быть снабжены достаточно полной документацией, обеспечивающей длительный (10—20 лет) период эксплуатации, сопровождения и развития.

ПРОМЕТЕЙ-технологии поддерживают методические материалы и инструментальные средства автоматизации технологического процесса. Последние обеспечивают возможность разработки программ для различных типов ЭВМ на технологической базе универсальных ЭВМ БЭСМ-6, ЕС ЭВМ и профессиональных персональных ЭВМ. Обеспечен контроль выполнения технологических этапов и качества разработки компонент и комплекса программ в целом. ПРОМЕТЕЙ-технология базируется на следующих принципах:

применение системы языков проектирования программ, поддерживающих все этапы технологического процесса (языки подготовки средств автоматизации, диалога, спецификаций, программирования и отладки);

использование системы взаимосвязанных языков программирования нескольких уровней;

применение программ, настраиваемых на характеристики проектируемых комплексов и ЭВМ инструментальных кросс-систем, базирующихся на больших универсальных и персональных ЭВМ;

раздельная компиляция и комплексирование модулей, написанных на языках программирования разных уровней, с использованием средств автоматизации и развитой системы библиотек проекта;

планируемое тестирование модулей, групп и комплексов программ с оценкой полноты выполненного тестирования;

обеспечение автоматизированной комплексной отладки программ в реальном масштабе времени;

выпуск полного комплекта программной документации и машинных носителей в соответствии с требованиями стандартов и особенностями реализующих ЭВМ.

Основные функции системы ПРА. Технологическая система ПРА разработана как автономная настраиваемая кросс-система. Система поддерживает ПРОМЕТЕЙ-технологии на базе профессиональных персональных ЭВМ с системами команд *i*8080 (КР580) и РДР-11 («Электроника 60»). Технической базой разработки

двух базовых версий ПРА служили микроЭВМ СМ-1800 (КР580) [2] и «Электроника НЦ-80-20/2» (ДВК-2) [3]. Обе базовые версии имеют одинаковые функции и структуру. Для управления системой ПРА используется небольшая специализированная операционная система (монитор) собственной разработки. Это позволило эффективнее использовать ограниченные ресурсы ПЭВМ и обеспечить разработку программ объемом до 20К байт. Система ПРА функционально и программно совместима с мощными кросс-системами ЯУЗА-6, ТЕМП, РУЗА [4], благодаря унификации функций, языков проектирования, технологических принципов создания программ при применении таких кросс-систем.

В соответствии с концепциями ПРОМЕТЕЙ-технологии языки проектирования программ построены на русской лексической базе и практически идентичны во всех технологических системах, входящих в ПРОМЕТЕЙ-технологии. Описания глобальных переменных подготавливаются и транслируются независимо от программных модулей. В качестве языка программирования используется базовый автокод со средствами макrorасширения, настраиваемый на различные типы микроЭВМ. Реализация с языка высокого уровня затруднена из-за ограниченных ресурсов памяти используемых ПЭВМ.

Разработанные версии системы ПРА автоматизируют следующие функции:

подготовку технологических средств в соответствии с характеристиками создаваемого комплекса программ и настройку системы на технические параметры микроЭВМ, для которой разрабатываются программы;

ввод, просмотр и редактирование текстов программ и описаний переменных на исходном языке программирования на экране дисплея;

управление базой данных проектирования (хранение, копирование и удаление данных, уплотнение библиотек, а также выдача на экран дисплея исходных текстов программ и описаний переменных);

трансляцию исходных текстов программ в объектный код различных реализующих ЭВМ;

тестирование и отладку модулей и групп программ в режиме интерпретации реализующей ЭВМ;

выдачу комплекта программной документации;

подготовку перфоленг для ввода программ в объектном коде.

Для освоения системы предусмотрены средства обучения, обеспечивающие выдачу справочной информации. Иерархическая система позволяет использовать справочную информацию по мере необходимости и не мешает специалистам, хорошо осваившим систему. Диагностические сообщения при выявлении ошибок и аномалии в реализации заданий имеют

единую структуру и, как все языки проектирования, основаны на русской лексике.

С использованием базовых версий кросс-систем ПРА могут быть подготовлены рабочие версии, настроенные на специфические системы команд и характеристики проектируемых комплексов программ. Для описания условий применения и характеристик настройки ЭВМ применяется специальный язык ФОН [4], имеющий три раздела. В первом — формально описываются характеристики реализующей ЭВМ: структура и особенности адресации памяти, форма представления и способ расположения данных, система и форматы команд, время выполнения операций, способы вычисления контрольных разрядов и т. д. Во втором — размещается описание автокода (ассемблера). На основе унифицированной лексики, синтаксиса и обобщенного формата базового автокода настройщик системы ПРА разрабатывает конкретное описание автокода реализующей ЭВМ. В третьем — располагается описание распределения ресурсов реализующей ЭВМ под различные компоненты проектируемого комплекса программ (глобальные и локальные переменные, константы, стандартные и обменные программы и т. д.). Здесь же закрепляются соглашения о связях и использовании различных регистров.

Описание условий применения рабочей версии системы ПРА размещается в базе данных настройки и автоматически транслируется в совокупность параметрических таблиц, которые размещаются в соответствующих местах базовой версии, превращая ее в настроенную рабочую версию. В рабочей версии только интерпретатор реализующей ЭВМ подготавливается неавтоматизированно, путем программирования каждой интерпретируемой команды.

Для работы пользователя в режиме диалога с ПЭВМ используется экранный редактор. Язык диалога использует русскую лексику и аналогичен языку кросс-системы РУЗА. Под экраном разбивается на три зоны: *зону директив* (строка 1), *зону текстов программ и данных* (строки 2..23), *зону диагностики и результатов выполнения заданий* (строка 24). Редактор текста позволяет вызывать, просматривать и корректировать текст, исключать и заменять символы путем прямого указания места курсором.

Для эффективного использования внешней памяти ПЭВМ исходные тексты программ и описания данных хранятся в библиотеке символьной информации в упакованном виде. При вызове текстов на экран дисплея осуществляется распаковка информации, так же как при выдаче на печать. Соответственно при пересылке с экрана в архив данные упаковываются. Библиотека исходных текстов катало-

гизирована, имеется возможность уплотнения и копирования по именам программ и описаний данных.

Реализована трансляция программ с автокода и макроязыка. Особенностью является возможность независимой трансляции описаний глобальных переменных, что характерно для некоторых языков высокого уровня. Это облегчает комплексирование групп программных модулей и уменьшает число ошибок в междомодульных связях. Трансляция производится в пять этапов (макрогенерация, лексический контроль, синтаксический контроль, распределение памяти переменных, генерация объектного кода) с выдачей результатов диагностики при завершении каждого этапа. Разветвленный лексический, синтаксический и семантический контроль обеспечивают свыше 50 диагностических сообщений об ошибках. При выявлении ошибки на любом шаге трансляция прекращается.

Отладка программ ведется на уровне языка программирования (автокod) методом детерминированного тестирования с использованием интерпретации. Для составления задания на тестирование и информирование о результатах исполнения отлаживаемой программы применяется язык отладки, позволяющий задавать имя отлаживаемого модуля, исходные значения регистров, переменных и элементов массивов, точки пуска программ и контроля исполнения, а также управлять выполнением задания. После остановки в контрольной точке можно вывести информацию о состоянии программы, изменить значения регистров или переменных, продолжить исполнение задания с любой метки отлаживаемой программы. Для выдачи на экран результатов отладки на уровне языка программирования используется паспорт отлаживаемой программы, подготавливаемый в процессе трансляции. В паспортах модулей содержатся описания меток, локальных и глобальных переменных, а также элементов массивов, что обеспечивает отображение основных результатов тестирования в символьном виде. При отладке взаимосвязанных программ необходимо задать их имена и последовательность вызова.

В связи с недостатком памяти выдача документации осуществляется сразу после трансляции. Если при выдаче документа транслятор обнаруживает ошибку, то документ не выдается. Для определения типа ошибки необходимо протранслировать программу непосредственно вызовом транслятора. Особенностью выдачи программной документации является возможность задания формы бланков с использованием «меток». При этом можно учесть особенности структуры системы команд реализующей ЭВМ и применяемых стандартов.

Программы в объектных кодах для вывода в реализующую ЭВМ записываются

на перфокартах с учетом требований соответствующих устройств ввода. Обеспечена возможность формирования новой структуры перфокарты при изменении видов микросхем памяти микроЭВМ.

Конструктивные особенности системы ПРА. Система ПРА включает ряд подсистем, решающих частные задачи (рис. 1):

монитор (МОН) обеспечивает управление подсистемами и взаимодействие с пользователем;

управление базой данных (УБД) осуществляет функции накопления и изменения данных о программах и переменных в библиотеках проектирования;

редактор текста (РЕД) позволяет выводить на экран и редактировать исходные тексты программ, описания данных и комментарии, а также выводить данные на печать или перфокарты;

транслятор (ТАК) обеспечивает трансляцию описаний данных, макрокоманд и программ на автокоде в объектный код реализующей ЭВМ;

подсистема детерминированного тестирования (ДТП) предназначена для отладки программ в режиме интерпретации;

подсистема документирования (ТДО) позволяет выпускать программную

документацию в заданных форматах; для вывода программ на перфокарты в объектных кодах в различной кодировке и ее контроля служит подсистема МАН;

трансляция описаний ЭВМ и условий применения всей системы осуществляется подсистемой настройки НАС.

Для эффективного использования оперативной (64К байта) и внешней (2×256К байт) памяти ПЭВМ минимизирована резидентная часть системы ПРА (подсистема МОН — 4К байт). Кроме того, строгое соблюдение модульности построения и динамической структуры загружаемых частей подсистем обеспечили полное использование оперативной памяти программами и данными только при исполнении соответствующих функций.

Программы подсистем располагаются на трех сторонах сменных дискетов (рис. 2), на каждой из которых имеются подсистемы МОН, УБД и РЕД. Остальные подсистемы сгруппированы на сторонах дискетов в соответствии с технологическими этапами работ: для разработки программ — ТАК и ДТП; для выпуска документов и машинных носителей — ТДО и МАН; для настройки системы на условия применения — НАС. При

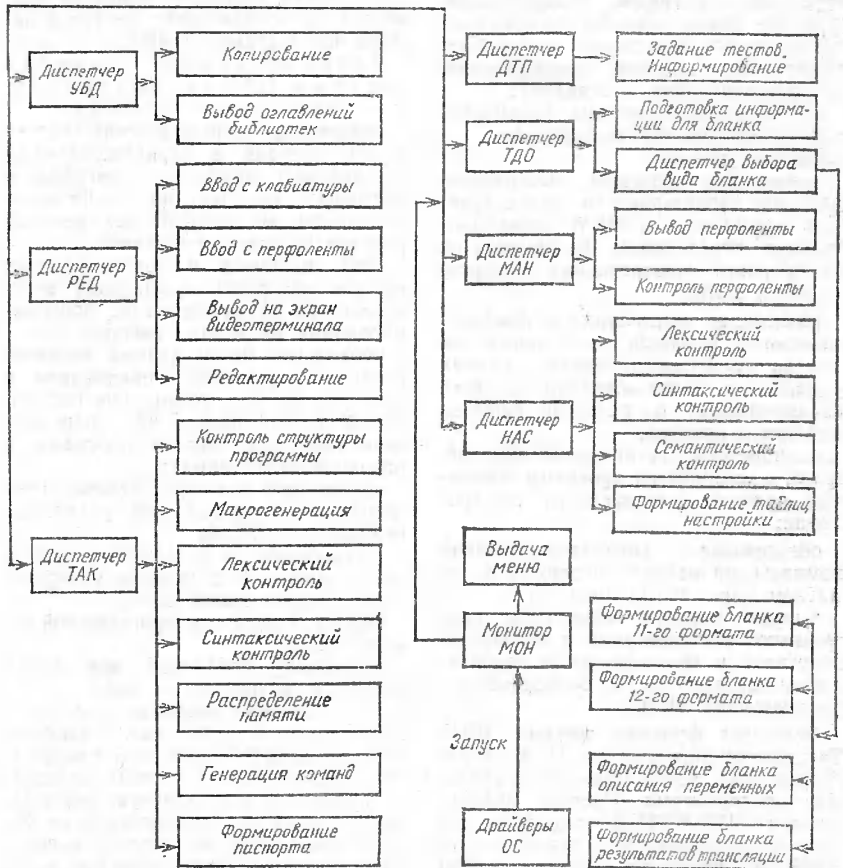


Рис. 1. Структурная схема системы ПРА

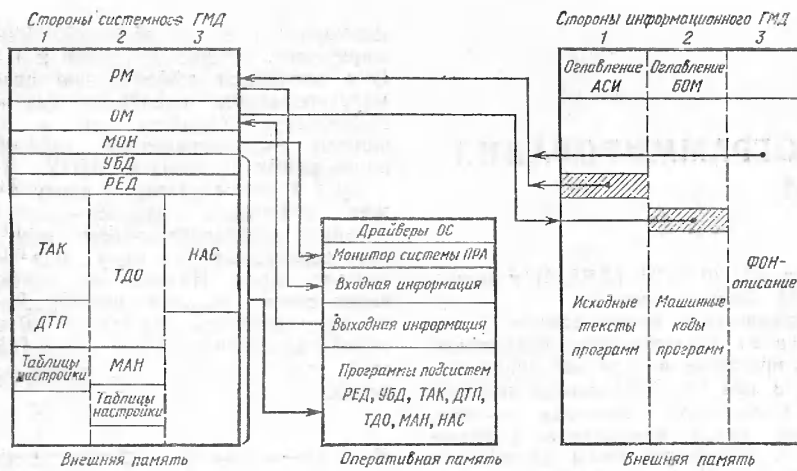


Рис. 2. Ресурсы памяти системы ПРА

изменении этапа работ дискетта устанавливается в дисковде соответствующей стороной.

Для улучшения использования ресурсов ПЭВМ по памяти и производительности штатные операционные системы ЭВМ СМ-1800 и ДВК-2 почти полностью удалены из внешней памяти. Сохранены только компоненты ОС, размещенные в постоянном запоминающем устройстве (4К байт), обеспечивающие инициализацию системы ПРА, и некоторые драйверы.

База данных проектирования размещается на трех (или более) сторонах дискетт и включает:

архив символьной информации (АСИ), предназначенный для хранения исходных текстов программ и описаний переменных (1-я сторона) до 150 модулей;

библиотеку до 150 объектных модулей (БОМ), содержащую программы в машинных кодах (2-я сторона); библиотеку описания условий применения (3-я сторона).

Если исходные тексты программ не размещаются на одной стороне дискетты, то допускается последовательное использование нескольких сторон. Для повышения надежности хранения данных, сокращения числа обращений к базе данных пользователя и уменьшения времени реакции системы вся работа с конкретной программой ведется в системном наборе данных (PM). Для этого программа предварительно копируется средствами системы ПРА из библиотеки исходных текстов или вводится с экраны или перфоленты. При этом обеспечивается возможность просмотра и редактирования текста на экране, трансляция, вывод на печать или на перфоленту. Программа в объектном коде и ее паспорт, сформированные транслятором, первоначально размещаются в системном наборе данных (OM) и могут быть использованы для отладки с применением системы

ДТП. По инициативе пользователя данные системных наборов копируются соответственно в АСИ и БОМ. При копировании программы обеспечивается размещение ее на место, занимаемое до модификации, если места достаточно, или в конце библиотеки. При отсутствии места, достаточного для размещения новой или модифицированной программы, производится автоматическое уплотнение библиотеки. В библиотеку объектных модулей оттранслированные программы размещаются по начальным адресам, указанным пользователем.

Необходимость замены рабочей стороны дискетт, хранящих программы или данные, подсказывается диагностическими сообщениями на экране дисплея. Для развития системы предусмотрена возможность увеличения используемого числа сторон дискетт, а также размещение программ и данных во внешней памяти на

Таблица

Оценка переносимости программ системы ПРА

Название подсистемы	Объем подсистемы, К байт	Объем машинно-зависимой части, К байт	Доля машинно-зависимости, %
MON	2,3	2,3	100
УБД	2,3	2,3	100
РЕД	9,4	9,4	100
ТАК	51,1	11	20
ДТП	15,6	5,8	37
ТДО	28,1	4	14
МАН	8,7	8,7	100
НАС	49,8	8,4	17
Итого	167,3	51,4	30

винчестерских дисках емкостью 10... 20 Мбайт.

Разработка подсистем ТАК, ДТП и ТДО производилась в значительной части на ЭВМ БЭСМ-6 с использованием кросс-системы ТЕМП [5]. Исходные тексты модулей готовились на макроязыке и транслировались сначала в тексты ассемблера, а далее в объектные коды ПЭВМ СМ-1800 и ДВК-2. Первоначально модули отлаживались на ЭВМ БЭСМ-6 в режиме интерпретации системы команд СМ-1800. Основная часть программных модулей (70 %) базовых версий системы ПРА имеет идентичные исходные тексты, что позволило значительно сократить затраты на отладку (см. таблицу). Подсистемы MON, UBD, RED, MAH разрабатывались на ассемблере соответствующих ПЭВМ с использованием их штатных операционных систем. На ДВК-2 создана версия системы ПРА для разработки программ микроЭВМ, реализуемых на микропроцессоре i8086 [6]. Версия включает интерпретатор системы команд i8086 с некоторыми ограничениями. Система ПРА передана на изготовление и хранение в Центрпрограммсистем.

ЛИТЕРАТУРА

1. Технология проектирования комплексов программ АСУ/В. В. Липаев, Л. А. Серебровский, П. Г. Гаганов и др.— М.: Радио и связь, 1983.— 264 с.
2. Гиглавы А. В., Кабанов Н. Д., Прохоров Н. Л., Шкамарда А. Н. МикроЭВМ СМ-1800.— М.: Финансы и статистика, 1984.— 136 с.
3. Глухман В. Л., Хохлов М. М., Лозовой И. О. Архитектура и программное обеспечение диалоговых вычислительных комплексов «Электроника НЦ-80-20». Всесоюзная научно-техническая конференция «Диалог-82-микро», Тез. докладов, 1982, с. 17—23.
4. Технология проектирования комплексов программ АСУ/В. В. Липаев, Л. А. Серебровский, П. Г. Гаганов и др./Под ред. Ю. В. Асафьева и В. В. Липаева.— М.: Радио и связь, 1982.— 360 с.
5. Липасв В. В., Каганов Ф. А. Система автоматизации технологии разработки комплексов управляющих программ для микропроцессоров и микроЭВМ (ТЕМП).— Управляющие системы и машины, 1980, № 1, с. 32—36.
6. Уокерли Дж. Архитектура и программирование микроЭВМ: Пер. с англ./Под ред. А. Г. Филиппова. Т. 2.— М.: Мир, 1984.— 341 с.

Статья поступила 21 мая 1984 г.

И. В. Вельбицкий, А. Л. Ковалев

ГРАФИЧЕСКИЙ СТИЛЬ ПРОГРАММИРОВАНИЯ ДЛЯ ПЕРСОНАЛЬНОЙ ЭВМ

Появление и широкомасштабное внедрение персональных ЭВМ (ПЭВМ), как нового «инструмента формализации профессиональных знаний», потребовало создания качественно новой технологии программирования, рассчитанной в первую очередь на функциональных специалистов (физиков, математиков, биологов, инженеров, работников аппарата управления и т. д.).

Имея конечной целью быструю разработку качественных программ, технология программирования ПЭВМ должна обеспечивать максимальное упрощение и облегчение процедур их создания.

Для большинства пользователей ПЭВМ является первой вычислительной машиной, с которой они сталкиваются, и важно с самого начала работы прививать навыки дисциплинированного программирования. Учитывая то, что у этой категории пользователей практически нет возможностей для глубокого изучения основ программирования, предлагаемая им методология должна быть доведена до такого уровня, чтобы будучи реализованной в технологической системе, стать средством, полностью определяющим дисциплину работы программиста и гарантировать заданный уровень технологической культуры программирования и качества разрабатываемых программ.

Принцип активности технологии должен проявляться и через такие характерные для режима персональных вычислений особенности, как обеспечение «дружественного» интерфейса, занимательное преподнесение материала, широкое использование графических средств отображения информации.

В этом отношении возможности современной техники все явственнее вступают в противоречие со старыми методами представления программ, что диктует необходимость поиска новых форм их представления, ориентированных в первую очередь на облегчение зрительного восприятия.

Предпосылки для реализации этих принципов содержит разработанная в Институте кибернетики имени В. М. Глушкова АН УССР Р-технология программирования. Ниже рассматривается инструментальный комплекс поддержки Р-технологии — РТК-микро на персональной ЭВМ «Электроника 80-20/2» (ДВК-2) и микро-ЭВМ «Электроника 60».

Графическая форма записи. В отличие от традиционного представления программ в виде линейного текста в основу Р-технологии положена графическая знаковая система, более емкая, компактная, а также лучше соответствующая ассоциативному принципу обработки информации. В Р-технологии предлагается не писать, а рисовать программу в языке нагруженных по дугам графов.

Нагруженный по дугам ориентированный граф, изображенный с помощью только вертикальных и горизонтальных линий и состоящий из структур (подграфов), каждая из которых имеет только один вход и один выход, называется в Р-технологии согласно ГОСТ 19.005—85 Р-схемой. Для ввода Р-схемы в среднем требуется в 1,5 раза меньше нажатий на клавиатуру дисплея, чем для ввода операторов в традиционной текстовой форме (рис. 1).

Р-схемы могут использоваться для изображения алгоритмов, программ, данных, процессов, схем — функциональных, информационно-логических,

а) if A>B
then A:=A-B
else B:=B-A

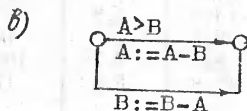
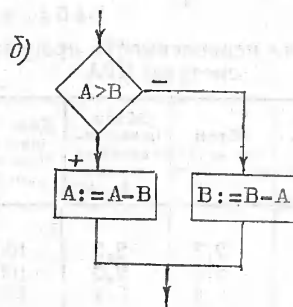


Рис. 1. Различные формы записи условного оператора: а — традиционная текстовая форма, б — блок-схема, в — Р-схема.

поточковых, а также технологических маршрутов, сетевых графиков и т. д. Они допускают эффективную автоматизированную поддержку как на графических дисплеях, так и на широко распространенных алфавитно-цифровых дисплеях и АЦПУ.

Дуга Р-схемы сверху и снизу может содержать надписи — произвольную последовательность символов, записанных в одну или несколько строк. Надпись над дугой имеет смысл условия прохождения по дуге. Надпись под дугой имеет смысл действия, которое выполняется, если условие над дугой истинно.

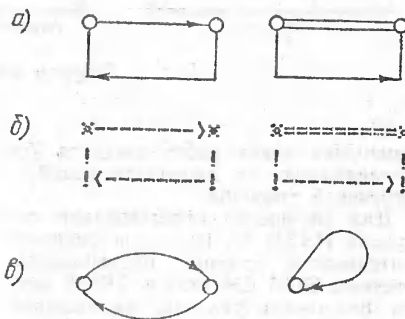


Рис. 2. Базовые графические структуры

Р-схемы в комплексе РТК-микро строятся из двух базовых графических структур (рис. 2,а), а соответствующее изображение — на существующих алфавитно-цифровых дисплеях и АЦПУ (рис. 2,б), причем число дуг в каждой структуре может быть произвольным. Такой способ изображения адекватен графам, приведенным на рис. 2,в и удобнее для написания текста на дугах. Две параллельные линии между вершинами (выполненные на рис. 2,б символом «=») означают отождествление двух вершин Р-схемы. Если из вершины Р-схемы исходит несколько дуг, то переход осуществляется по первой сверху, у которой истинно условие, записанное над дугой.

Над базовыми структурами в комплексе РТК-микро определены две операции их соединения: последовательное (рис. 3) и вложенное (рис. 4). В результате соединения базовых структур с помощью указанных операций получают структуры Р-схемы, над которыми также возможны эти операции. Вложенное соединение структур осуществляется путем замены дуги структуры на последовательное соединение этой дуги и вкладываемой структуры.

Ввод в ЭВМ и модификация графических программ осуществляется с помощью специального графического редактора. Редактор позволяет вводить графические структуры, соединяя их последовательно или

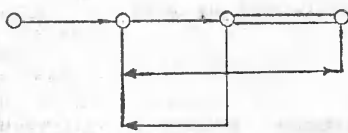


Рис. 3. Последовательное соединение графических структур

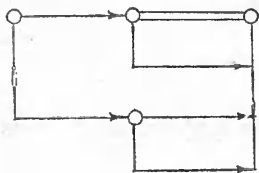


Рис. 4. Вложенное соединение графических структур

вкладывая друг в друга; добавлять новые дуги в уже существующие структуры; переставлять структуры в графе; удалять отдельные дуги, структуры или полностью весь граф, вводить и модифицировать текстовые строки на графе и вне графа.

Ввод и редактирование текста, а также передвижение маркера по экрану выполняется с помощью алфавитно-цифровой клавиатуры и специальных функциональных клавиш дисплея. Ввод и редактирование графа построено на основе меню пиктограмм, элементы которого располагаются в нижней строке экрана. Ввод графической структуры сводится к указанию специальным маркером нужной пиктограммы и нажатия клавиши исполнения, что приводит к автоматическому «рисованию» структуры на экране дисплея. Маркер в графе всегда находится в пределах какой-либо структуры и расположен на конкретной ее дуге. Эти структура и дуга являются текущими. Новую структуру в графе можно ввести слева или справа от текущей (последовательное соединение) или вложив в текущую дугу (вложенное соединение). Новые дуги в структуре добавляются выше или ниже текущей дуги. С помощью соответствующих команд меню могут быть удалены текущая дуга, структура, граф.

Используя редактор, можно осуществить сборку программы требуемой конфигурации на экране дисплея из базовых графических структур графа. Такое наглядное конструирование наряду с упрощением программирования повышает его занимательность, внося характерный для ПЭВМ элемент игры. Исходный текст графической программы хранится в виде файла, содержащего текстовые строки и управляющие коды, соответствующие графическим структурам и дугам, и разворачивается в графическое изображение при выводе на экран или на печать. Для программирования комплексов

```

a) PROGRAM PRINTID;
   VAR
     C:CHAR#;
   BEGIN
     READ(C);
     WHILE C('>') DO
       IF C='>' THEN
         BEGIN
           WRITELN;
           READ(C);
           WHILE C='>' DO
             READ(C);
         END
       ELSE
         BEGIN
           WRITE(C);
           READ(C);
         END;
       WRITELN;
     END.
  
```

```

b) PROGRAM PRINTID;
   VAR C:CHAR#;
   *-----*X=====*-----*X-----*
   READ(C);C('>') C='>' !WRITELN
   !-----*X=====*-----*
   !WRITELN!C='>' !
   !READ(C)!----->!
   ! READ(C)!
   !----->!
   WRITE(C)
   READ(C)
  
```

Рис. 5. Программа на языке Паскаль (а) и графический Паскаль (б)

РТК-микро предлагает графические языки: Паскаль, Фортран, Си, ассемблер.

Описательная часть программы в языке графический Паскаль совпадает с записью описательной части программы в традиционном Паскале (рис. 5, а, б). Исполнительные части программы, процедур и функций задаются отдельными графами. Весь текст программы, учитывая вложенные процедуры и функции, может содержать несколько графов. Над дугой графа в качестве условия прохождения по дуге записывается логическое выражение. Под дугой могут находиться операторы присваивания и вызова процедуры. Каждый оператор записывается в отдельную строку, но при необходимости может занимать несколько строк. Разделитель «;» между операторами не ставится. На рис. 6 представлен пример программы, состоящей из двух графов. Аналогично реализованы языки графический Фортран (рис. 7) и графический Си.

На рис. 8 приведен пример программы на графическом ассемблере. Исполнительная часть программы также задается с помощью графов. Первый граф является головным, к остальным графам происходит обращение как к внутренним подпрограммам. Допускается несколько входов в программу. Условие над дугой состоит из последовательности операторов макроассемблера, заканчивающейся мнемокодом проверки условия, соответствующим мнемокодам операторов ветвления макроассемблера: EQ, NE, LT, GE и т. д. Другая форма записи условия содержит мнемокод сравнения, за которым следуют два сравниваемых операнда. Под дугой записываются любые операторы макроассемблера за исключением операторов ветвления. Метки допускается использовать только в разделе описаний и в качестве имени всего графа.

Трансляторы с графических языков реализованы как препроцессоры

к трансляторам с соответствующих базовых языков, входящих в состав программного обеспечения штатной операционной системы. Ориентированный граф в графических языках задает управляющую структуру программы и заменяет собой использование разнообразных операторов управления if, while, repeat, for и т. п., записанных в ненаглядной текстовой форме. При этом отпадает необходимость использования ключевых слов if, while, ..., операторных скобок begin-end, большинства разделителей, меток. На дугах графических структур для всех процедурно-ориентированных языков от Фортрана до АДЫ записываются одни и те же функциональные элементы — логические выражения, операторы присваивания и операторы обращения к процедурам. Эти элементы для большинства языков имеют стандартный (унифицированный) синтаксис, упрощающий программирование. Графическая форма записи является единой для всех языков, что улучшает понимаемость программ, облегчает освоение языка программирования и переход от одного языка к другому. Так как сборка графа программы осуществляется редактором из законченных графических структур, P-схема как графическая структура остается всегда правильной, благодаря чему исключается целый класс ошибок, связанных с неверной передачей управления, несоответствием операторных скобок begin-end или ключевых слов в операторах управления, например if-then-else. Отсутствие целого ряда ставших ненужными ключевых слов и разделителей позволяет уменьшить исходный текст графической программы вдвое по сравнению с текстом в традиционной линейной форме записи.

P-технология автоматически реализует структурный стиль программирования [1], так как он заложен непосредственно в ее знаковую систему. Одновременно обеспечивается более высокий уровень структурности записи программы за счет использования для задания логики программы (наиболее важной и

В обеспечении высокого качества результирующих программ при общем упрощении процедуры разработки проявляется принцип активности Р-технологии. Вся работа с комплексом РТК-микро построена на основе использования меню, которое скрывает от пользователя ненужные детали инструментальных средств Р-технологии и соответствующей операционной системы, повышая технологичность связи с функциональным специалистом.

Проектирование в неформальных обозначениях. Дальнейшее развитие графический стиль программирования, а также принцип активности технологии находят в предлагаемых Р-технологией средствах проектирования в неформальных обозначениях, позволяющих совместить процессы разработки алгоритмов, программных средств и качественной документации. При этом документация — вторичный продукт этапа проектирования в Р-технологии, получаемый с минимальными дополнительными затратами.

Средства проектирования обеспечивают разработку программ методом нисходящего пошагового уточнения с оформлением каждого модуля проектирования в некоторой стандартной форме, получившей название чертежа [3]. Каждый чертеж разделяется на три поля — поле спецификаций, рабочее поле и поле абстракций. Соответственно процесс проектирования на каждом уровне распадается на три четко выраженных этапа (рис. 9).

Поле спецификаций содержит неформальное описание задачи, решаемой на данном уровне проектирования. Описание происходит в процессе ответов на предлагаемую в диалоге анкету, содержащую ряд обязательных пунктов — цель работы, входные данные, выходные данные, описание алгоритма и ряд дополнительных пунктов, включаемых при необходимости; условия применения, особые случаи, требования к оборудованию и т. п. В процессе такого описания происходит концентрация внимания разработчика на важных сторонах решаемой задачи до ее формального проектирования и выполняется документирование данного уровня проекта в единой для всех чертежей проекта форме.

Рабочее поле содержит формальное описание алгоритма данного уровня проектирования в графическом языке, причем на дугах графа и в описательной части наряду с предложениями языка программирования допускается использование произвольных обозначений, наиболее удобных в зависимости от характера проектируемой задачи — фраз на естественном языке, формульного текста, специальных обозначений. Это еще больше повышает наглядность оформления проекта.

Поле абстракций содержит определения использованных в рабочем поле неформальных обозначений. Определение осуществляется следующим способом. В диалоговом режиме на экран последовательно

выводятся использованные в рабочем поле обозначения, для которых либо вводится эквивалент в формальном языке (Паскаль, макро-ассемблер и т. д.), либо определенные обозначения переносятся на следующий уровень проектирования и реализуется в дальнейшем отдельным чертежом.

Чертеж, таким образом, выступает как важное многоплановое понятие. Во-первых, это программная единица — модуль проекта. Во-вторых — единица документации. Ценность документации заключается в том, что она составляется до написания программы, рассредоточена по дереву проекта и привязана к конкретным уровням проектных решений. В третьих — это единая технологическая схема выполнения процесса проектирования, заставляющая четко ставить начальную задачу до реализации каждого модуля и оговаривать его интерфейсы, предлагающая для описания алгоритма наиболее адекватные средства, включая записи на естественном языке, позволяющая откладывать детали реализации алгоритма на более поздние этапы проектирования.

Реализуемый проект во внешней памяти представляется двумя файлами — файлом, содержащим каталог проекта, и файлом чертежей. Проектирование осуществляется в диалоге с помощью процессора проектирования, управляемого посредством меню. Название головного чертежа вносится в каталог по запросу процессора. Названия остальных чертежей поступают в каталог из поля абстракций автоматически. Допускается несколько ссылок на один и тот же чертеж, что позволяет использовать при проектировании стандартные модули.

Проектирование заканчивается, когда все использованные при описании проекта обозначения доведены до формальной реализации. После этого выполняется процедура сборки, формирующая готовую программу на формальном языке. Допускается сборка и незавершенного проекта; в этом случае нереализованные чертежи заменяются автоматически генерируемыми заглушками. На этапе отладки может быть получена трасса выполнения программы в терминах, использованных при проектировании обозначений.

Проектная документация на программные средства состоит из каталога и комплекта чертежей, который может быть выведен на печать полностью либо частично. Оформленные документации в виде чертежей в неформальных обозначениях облегчает сопровождение программного проекта (внесение в него изменений, улучшений, исправление ошибок, обнаруженных в процессе эксплуатации, и т. п.), обмен алгоритмами, перенос с одной ЭВМ на дру-

```

-----
! РАСЧЕТ ЗАТРАТ МАШИННОГО ВРЕМЕНИ
! ----- ПОЛЕ СПЕЦИФИКАЦИЯ -----
! ЦЕЛЬ: РАСЧЕТ ПО ЗАПРОСУ ЗАТРАТ МАШИННОГО ВРЕМЕНИ И ВЫВОД
! РЕЗУЛЬТАТОВ НА ПЕЧАТЬ
! ВХОД: БАЗА ДАННЫХ РАСХОДА МАШИННОГО ВРЕМЕНИ
! ВЫХОД: ТАБЛИЦА РАСХОДА МАШИННОГО ВРЕМЕНИ, ФОРМА ОПРЕДЕЛЯЕТСЯ
! В ДИАЛОГЕ С ПОЛЬЗОВАТЕЛЕМ
! АЛГОРИТМ: ЗАПРОС ИМЕНИ БА И ЕЕ ОТКРЫТИЕ, ЕСЛИ БА НЕ НАЙДЕНА,
! ЗАПРОС ПОВТОРЯЕТСЯ; ЗАПРОС ДАННЫХ О ФОРМЕ РАСЧЕТА;
! РАСЧЕТ ЗАТРАТ МАШИННОГО ВРЕМЕНИ И ВЫВОД РЕЗУЛЬТАТА НА
! ПЕЧАТЬ В ВИДЕ ТАБЛИЦЫ
! ----- РАБОЧЕЕ ПОЛЕ -----
! ОПИСАНИЕ ПЕРЕМЕННЫХ
!
! *-----*)
! 1 ЗАПРОС ИМЕНИ БА 1 ЗАПРОС О ФОРМЕ РАСЧЕТА
! 1 1 РАСЧЕТ РАСХОДА МАШИННОГО ВРЕМЕНИ
! 1 БА НЕ НАЙДЕНА 1 ПЕЧАТЬ РЕЗУЛЬТИРУЮЩЕЙ ТАБЛИЦЫ
! 1 <-----!
!
! ----- ПОЛЕ АБСТРАКЦИЯ -----
! 3 ОПИСАНИЕ ПЕРЕМЕННЫХ
! VAR W:DEFREL;
! NM:ALFA;
! 3 ЗАПРОС ИМЕНИ БА
! WRITE ('ИМЯ БА? ')
! READLN(NM)
! 3 БА НЕ НАЙДЕНА
! NOT INITDB(W,NM)
! 4 ЗАПРОС О ФОРМЕ РАСЧЕТА
! 4 РАСЧЕТ РАСХОДА МАШИННОГО ВРЕМЕНИ
! 4 ПЕЧАТЬ РЕЗУЛЬТИРУЮЩЕЙ ТАБЛИЦЫ
-----

```

Рис. 9. Пример оформления чертежа

гую, настройку «под ключ» по условиям эксплуатации и т. д.

Описанные средства являются единым инструментом для проектирования и разработки (кодирования) алгоритмов и программ. Такое единство инструментальных средств позволяет фактически стереть грань между отдельными этапами работ, а значит сделать технологию программирования более простой, естественной и эффективной.

Автоматизация управления разработкой программ. Может показаться, что для ПЭВМ не нужна организация управления разработкой программного средства. Однако одним из основных профессиональных применений ПЭВМ является использование в многомашиной технологической линии для промышленной разработки коллективом программистов отдельных модулей программного проекта. В этой технологической линии ПЭВМ используется как высокопроизводительный интеллектуальный терминал, подключенный в сеть больших ЭВМ. В этом случае средства управления разработкой программ на ПЭВМ являются частью общих средств автоматизации управления разработкой программного проекта. Но даже для чисто персональных применений средства управления разработкой на ПЭВМ необходимы для самоконтроля и обеспечения высокой технологической культуры проведения работ. Комплекс РТК-микро содержит в своем составе средства автоматизации управления разработкой программ, обеспечивающие планирование и контроль исполнения планов, контроль за состоянием проекта, учет людских и машинных затрат на разработку, получение количественных и качественных характеристик разработанных программ, расчет экономических показателей разработки.

Разработка программ в комплексе организована по безбумажной схеме. Это означает, что максимум циркулирующей в процессе разработки информации хранится в памяти самой ЭВМ и может обрабаты-

ваться системными средствами. Информационная база поддерживается с помощью входящей в состав технологического комплекса СУБД реляционного типа, основной особенностью которой является простой и наглядный экранный интерфейс, позволяющий использовать ее в качестве информационно-справочной системы для получения всевозможных сведений об интегральном состоянии проводимых коллективом программистов (или отдельным программистом) работ. На рис. 10 показан пример выведенной на экран базы данных. СУБД дополняется генератором отчетов, позволяющим получить сведения из информационной базы в виде документов, печатаемых по заранее заготовленным шаблонам. На рис. 11 показана структура средств автоматизации управления разработкой программ.

Подсистема планирования автоматизирует машинное ведение планов, опираясь на базу данных «План», которая содержит сведения о проводимых коллективом программистов работах: названия работ, фамилии исполнителей, типы работ, плановые и фактические сроки выполнения как в целом работ, так и отдельных подэтапов, текущее состояние работ, при выполнении работы несколькими исполнителями процент участия каждого и другие сведения. При вводе планы проверяются на полноту (должны быть учтены все члены коллектива на всем интервале планирования) и корректность (правильность установки сроков работ). При этом дополнительно используется база данных «Кадры», содержащая сведения о коллективе разработчиков. Подсистема предоставляет руководителю информацию о текущем состоянии работы, о загрузке каждого разработчика, о приближающихся сроках окончания работ, о срыве сроков выполнения работ.

Подсистема учета машинного времени накапливает в базе данных «Машинное время» сведения о машинных затратах по каждой работе. Подсистема дополнительно гото-

вит сводки о распределении затрат машинного времени за произвольный отчетный период по работам и разработчикам.

По окончании разработки программы выполняются ее метрические исследования, позволяющие получить ряд количественных показателей: длину исходного и рабочего кода, число модулей, средний размер модуля и другие и ряд показателей, в совокупности дающих представление о качестве разработанной программы: степень специфицированности, степень использования стандартных модулей и т. д. Эти сведения формируют базу данных «Программы». На рис. 12 приведен пример вывода в виде гистограмм статистических данных, характеризующих размеры графических модулей.

После этого выполняются расчеты экономических показателей. На основе сведений о затратах людских ресурсов (база данных «План») и «Кадры») и машинных ресурсов (база данных «Машинное время») рассчитываются стоимость разработки и производительность труда программиста. Учитывая объем программы, определяются себестоимость единицы программной продукции и удельные затраты машинного времени. Вычисляется ряд вторичных показателей — число выходов на ЭВМ, среднее время сеанса и др. Результаты вычислений заполняют базу данных «Программы» и выводятся в виде справок на печать.

Полученные сведения создают фундамент для научно-обоснованного нормирования труда программистов, эффективного планирования и организации производства программ с учетом возможностей каждого разработчика, предоставляют объективные критерии для организации соревнования внутри коллектива разработчиков и оценки роста индивидуальной квалификации каждого программиста.

СУБД и генератор отчетов позво-

БАЗА ДАННЫХ: KADRY : ЧИСЛО СТОЛБЦОВ: 5 : ЧИСЛО ЗАПИСЕЙ: 165					

N :	Ф.И.О.	ПОЛ :	ГОД РОЖД. :	ОКЛАД :	ТЕЛЕФОН
0 :	1	2 :	3	4 :	5
1:	ИВАНОВ И.И.	М	1950	180	66-25-13
2:	ПЕТРОВА П.П.	Ж	1955	165	
3:	СИДОРОВ С.С.	М	1940	220	27-01-15
			...		

Рис. 10. Пример вывода базы данных на экран

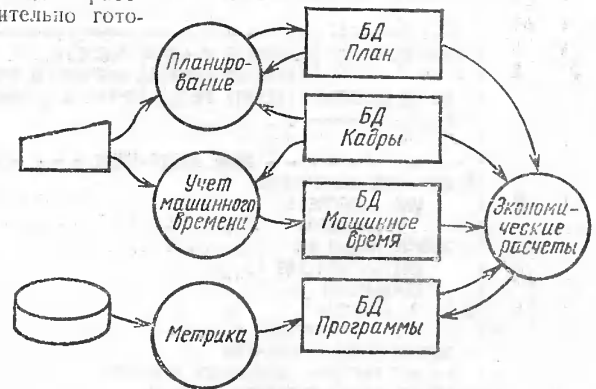


Рис. 11. Структура средств автоматизации управления разработкой программ

ЧИСЛО СТРУКТУР В ГРАФЕ
(ДАННЫЕ ПО 756 ГРАФАМ)

1	195	25.79%	*****
2	147	19.44%	*****
3	111	14.68%	*****
4	108	14.29%	*****
5	97	12.83%	*****
6	46	6.08%	*****
7	23	3.04%	*****
8	11	1.46%	***
9	8	1.06%	**
10	4	0.53%	*
11	3	0.40%	*
12	1	0.13%	*
13	1	0.13%	*
18	1	0.13%	*

ЧИСЛО ДУГ В ГРАФЕ
(ДАННЫЕ ПО 756 ГРАФАМ)

1	135	17.86%	*****
2	110	14.55%	*****
3	90	11.90%	*****
4	56	7.41%	*****
5	63	8.33%	*****
6	75	9.92%	*****
7	42	5.56%	*****
8	50	6.61%	*****
9	30	3.97%	*****
10	28	3.70%	*****
11	17	2.25%	*****
12	18	2.38%	*****
13	10	1.32%	***
14	11	1.46%	****
15	5	0.66%	**
16	5	0.66%	**
17	5	0.66%	**
18	2	0.26%	*
19	1	0.13%	*
21	2	0.26%	*
31	1	0.13%	*

РАЗМЕРЫ ГРАФА ПО ГОРИЗОНТАЛИ В ПОЗИЦИЯХ
(ДАННЫЕ ПО 756 ГРАФАМ)

< 11	16	2.12%	****
< 21	88	11.64%	*****
< 31	114	15.08%	*****
< 41	106	14.02%	*****
< 51	97	12.83%	*****
< 61	87	11.51%	*****
< 71	105	13.89%	*****
>= 71	143	18.92%	*****

РАЗМЕРЫ ГРАФА ПО ВЕРТИКАЛИ В СТРОКАХ
(ДАННЫЕ ПО 756 ГРАФАМ)

< 9	337	44.58%	*****
< 17	270	35.71%	*****
< 25	99	13.10%	*****
< 33	26	3.44%	****
< 41	15	1.98%	**
>= 41	9	1.19%	**

ляют оперативно в диалоге пере-
строить структуру базы данных и
шаблонов печатаемых документов.
Управление в комплексе РТК-микро
реализуется на основе меню, кото-
рые также могут в диалоге изме-
няться с помощью входящего в со-
став комплекса генератора меню.

Технологический комплекс РТК-
микро разработан в 1983 г. для опе-
рационных систем ОС ДВК и ФО-
ДОС (РАФОС), эксплуатируемых
соответственно на ЭВМ «Электроника
ИЦ-80-20» и «Электроника 60».
Комплекс реализован по Р-техноло-
гии на языке графический Паскаль
с использованием системы проекти-
рования в неформальных обозначе-
ниях. Накопленный опыт примене-
ния комплекса подтверждает его
высокую эффективность при разра-
ботке программ самого различного
назначения.

В настоящее время на Р-схемы
разработал ГОСТ 19.005—85 «Р-схе-
мы алгоритмов и программ. Обозна-
чения условные графические и пра-
вила выполнения» [4], который вхо-
дит в состав ЕСПД.

Помимо описанных в статье
средств комплекс РТК-микро вклю-
чает программы, реализующие обще-
системный технологический сервис:
экранный текстовый редактор, сред-
ства печати в несколько колонок,
программы оперативного копирова-
ния и контроля файлов, генерации
звуковых сообщений и др. Большин-
ство подсистем комплекса допускают
автономное использование, со-
здавая в то же время в совокупно-
сти интегральную систему с единой
идеологией. Комплекс РТК-микро
поставляется в виде дистрибутивных
дискет, содержащих инструменталь-
ные программы комплекса (около
600 К байт загрузочных модулей),
контрольные примеры и документа-
цию (около 400 К байт).

ЛИТЕРАТУРА

1. Дал О., Дейкстра Э., Хо-
ар А. Структурное программирова-
ние.— М.: Мир, 1975.— 248 с.
2. Ершов А. П. Персональная
ЭВМ—предок млекопитающих в
динозавровом мире ВЦКП.— В кн.:
Материалы Всесоюз. конф. «Диа-
лог-82-микро». Персональные ком-
пьютеры. Пушкино: ОНТИ НЦБИ
АН СССР, 1983, с. 9—25.
3. Вельбицкий И. В. Техно-
логия программирования.— Киев:
Техніка, 1984.— 279 с.
4. Вельбицкий И. В., Рай-
ков Л. Д., Маклаков А. В.,
Мкртумян А. А., Ушаков И. Б.
Новая графическая форма записи
алгоритмов и программ.— УСиМ,
1984, № 6, с. 51—56.

Рис. 12. Пример вывода на печать статистических данных о размерах графиче-
ских модулей

Статья поступила 6 августа 1985 г.

А. Т. Лолейт, С. И. Кочегарова

ОПЫТ ПРИМЕНЕНИЯ P-ТЕХНОЛОГИИ ПРИ РАЗРАБОТКЕ ПРОГРАММНЫХ СРЕДСТВ НА МИКРОЭВМ

Создание программных проектов на основе производственной технологии дает возможность получить программный продукт, отвечающий ряду наперед заданных требований. Наибольший интерес представляют технологии, минимизирующие затраты на сопровождение программных средств, составляющие 50 % стоимости программного проекта. В основу этих технологий положен принцип структурного программирования. Примером практической реализации указанного принципа являются технологии IPT, «Динамит» для ЕС ЭВМ и P-технология для ЕС ЭВМ, СМ ЭВМ и микроЭВМ.

P-технология — это способ организации разработки программ на базе графических языков конструкций, позволяющий получать структурные программы.

В Институте кибернетики имени академика В. М. Глушкова АН УССР разработаны три программных комплекса для различных поколений ЭВМ: РТК-ЕС, РТК-СМ, РТК-микро.

В отраслевом фонде алгоритмов и программ (ОФАП) проходил экспертизу и опытную эксплуатацию программный комплекс РТК-микро для ЭВМ «Электроника 100-25», «Электроника 60», диалогового вычислительного комплекса «Электроника НЦ-80-20/2М» (ДВК-2М).

Состав РТК-микро:

GRAFED — графический редактор;

GPA, GAS — преобразователи для языков Паскаль и ассемблер;

ED — текстовый экранный редактор;

PRINT — программа распечатки текстовых файлов,

PRINTG — программа распечатки файлов, содержащих графические конструкции;

RDWR — программа копирования файлов.

В процессе опытной эксплуатации программного комплекса РТК-микро на основе одного и того же алгоритма была разработана программа TXT, которая осуществляет распечатку документации в различных форматах традиционным методом (язык Паскаль) и с применением P-технологии (язык P-паскаль).

Время разработки программы на P-паскале по сравнению с обычным Паскалем уменьшилось примерно в 1,2 раза. Отметим, что программу с использованием P-технологии писал программист, ранее с P-паскалем

не знакомый. *Время отладки программы, написанной на P-паскале, сократилось приблизительно в четыре раза.*

Использование РТК-микро значительно экономит место на носителях данных, что особенно важно при существующих ограничениях на объем дисковой памяти для микроЭВМ «Электроника 60» и ДВК-2М (0,5 Мбайт). Экономия достигается за счет представления условных операторов и операторов цикла специальными символами, определяющими ветвление P-графа. *Объем исходных модулей снижается в два раза.*

Загрузочные модули, полученные трансляцией P-графа программы, имеют меньший объем (в 1,7 раза) благодаря использованию оптимизирующих свойств претранслятора GRA. Несмотря на то, что трудовые затраты на подготовку текста программы в виде P-графа с соответствующими комментариями несколько выше, чем подготовка текста на линейном Паскале, *общая трудоемкость разработки снижается на этапе разработки документации примерно в 1,6 раза.* Трудоемкий документ «Описание программы» может быть совмещен с документом «Текст программы». При этом в качестве блок-схемы алгоритма программы может быть использован P-граф программы с соответствующими комментариями.

Сопровождение программного средства, разработанного на базе РТК-микро значительно упрощается, так как модульность и иерархичность логических связей модулей позволяет программисту, не принимавшему участия в разработке данной программы, быстро локализовать ошибку и произвести модификацию, не затрагивая логических связей всего проекта в целом. *Трудоемкость на сопровождение уменьшается в 10 раз.*

По результатам опытной эксплуатации РТК-микро в ОФАП МСВТ следуют выводы:

1. РТК-микро может быть рекомендован широкому кругу пользователей микроЭВМ для разработки крупных программных проектов группами более двух человек;

2. РТК-микро может быть полезен для учета выработки программиста. Так как комплексе позволяет создавать программу из модулей ограниченной сложности, то выработка может учитываться величиной модуль/день;

3. К недостаткам РТК-микро следует отнести отсутствие графического отладчика, что вызывает необходимость использования листинга промежуточного файла типа PAS.

Комплекс РТК-микро может быть записан на отформатированные гибкие магнитные диски типа ИЗОТ 5257Е и поставлен пользователю вместе с технической документацией.

Статья поступила 8 апреля 1985 г.

Широко внедрять гибкие переналаживаемые производства и системы автоматизированного проектирования, автоматические линии, машины и оборудование со встроенными средствами микропроцессорной техники, многооперационные станки с числовым программным управлением, робототехнические, роторные и роторно-конвейерные комплексы.

Из проекта Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года

Для достижения поставленных перед народным хозяйством целей — быстрого роста эффективности общественного производства — необходимо в самое ближайшее время осуществить «реиндустриализацию» промышленных предприятий на базе широкого внедрения вычислительной техники, развитых систем связи, станков с числовым программным управлением, роботов и автоматизированных систем инженерного обеспечения производства.

При этом открываются качественно новые возможности повышения эффективности производства — организация работы предприятий на принципах гибкого автоматизированного производства (ГАП). Основным инструментом обеспечения гибкости современных производств являются мини- и микроЭВМ. Относительно низкая стоимость таких ЭВМ делает их доступными для широкого внедрения в промышленность, а универсальность, дополняемая высокой вычислительной мощностью, позволяет использовать их для автоматизации производства на всех его уровнях с возможностью быстрой перестройки на выпуск новых изделий.

На универсальном металлорежущем оборудовании традиционной структуры рабочий цикл составляет 10...15% общего фонда времени — остальное занимают подготовительно-заключительные операции. ГАП позволяют довести рабочий цикл до 70...80%. Таким образом производительность труда повышается в 2,5...5 раз.

В первых двух статьях предлагаемого цикла на примерах конкретных производственных систем рассматриваются принципы комплексной автоматизации машиностроительного производства. Материал третьей статьи посвящен одному из возможных подходов к программному обеспечению оперативного управления гибкими производственными системами. В последующих статьях рассматриваются вопросы, связанные с организацией управления основным компонентом гибкого машиностроительного производства — модулем «станок-робот», а также исполнительными устройствами технологического оборудования на базе микропроцессорной техники.

УДК 621.3.049.77.022.72

Е. О. Адамов, С. М. Дукарский, Д. А. Корягин

ИНТЕГРИРОВАННАЯ МАШИНОСТРОИТЕЛЬНАЯ СИСТЕМА КАПРИ

Одним из важнейших факторов ускорения внедрения результатов науки в народное хозяйство является совершенствование опытных производств новой техники на основе автоматизации инженерного обеспечения производства и применения программно-управляемого технологического оборудования.

Большой объем проектно-конструкторских и производственно-монтажных работ по созданию экспериментальных установок и стендов требует привлечения труда конструкторов, проектировщиков, технологов, рабочих различных специальностей. При этом выполняется до 70% нетворческих, рутинных операций, что делает эти профессии социально-непривлекательными и приводит к хроническим затруднениям с набором специалистов, текучести кадров.

В то же время, как показывает анализ этих

работ, при современном развитии вычислительной техники значительная часть рутинных операций может быть автоматизирована. Этим целям служит разработанная Институтом атомной энергии им. И. В. Курчатова совместно с Институтом прикладной математики им. М. В. Келдыша АН СССР при участии ряда организаций и предприятий система комплексной автоматизации проектирования, разработки и изготовления изделий машиностроительного профиля в опытных производствах НИИ и КБ (программа КАПРИ)*.

Основные социально-экономические цели программы КАПРИ:

* Адамов Е. О., Гнеденко В. Г. и другие. Интегрированная система автоматизированного проектирования и производства изделий опытным машиностроительным производством. — Вестник машиностроения, 1985, № 1, с. 34—40.

сокращение времени внедрения результатов науки в производство;

снижение трудовых затрат во всех звеньях цикла внедрения результатов науки в производство (проектирование, технологическая подготовка производства и производство — благодаря повышению производительности труда);

рост эффективности использования оборудования (увеличение коэффициентов технического использования и сменности);

сокращение доли нетворческих, трудоемких, утомительных работ;

снижение потребности в социально-непривлекательных профессиях, изменение качественной структуры производственных коллективов в пользу творческого умственного труда;

повышение качества и уровня стандартизации работ.

Предметная область программы КАПРИ — опытное машиностроительное производство, обеспечивающее создание крупных экспериментальных энергетических стендов. Характерные особенности такого производства:

индивидуальный и мелкосерийный масштаб (среднее число деталей одного наименования — до 10 единиц);

непродолжительное время обработки изделий (0,5...1 ч в зависимости от модели оборудования);

непродолжительное время подготовительно-заключительных и вспомогательных операций, составляющее примерно 50 % общего времени цикла обработки;

широкая номенклатура изделий (100 и более тысяч наименований в год);

выполнение конструкторских работ без предварительного макетирования;

неполный объем и упрощенные формы технологического проектирования;

нестабильная номенклатура материально-технического обеспечения.

Характер производства, а также отмеченные выше особенности деятельности инженерных подразделений, связанных с ним, требуют взаимосогласованного решения задач, возникающих в рамках работ по программе КАПРИ. Идеологическим стержнем программы является концепция интеграции технических средств машиностроительного производства и всех служб инженерного обеспечения в единую систему на основе развитого комплекса вычислительной техники и программного обеспечения.

Очевидными преимуществами такой интегрированной машиностроительной системы (ИМС) являются сокращение межоперационных трудовых затрат в циклах конструирования, технологической подготовки производства и изготовления изделий, внедрение безбу-

мажной технологии документооборота и сокращение аппаратной, программной и информационной избыточности.

В настоящее время в состав интегрированной машиностроительной системы КАПРИ включены следующие подсистемы:

автоматизации конструирования САПР-К;

автоматизации проектирования технологии САПР-Т;

организационно-экономического управления АСОЭУ;

оперативно-диспетчерского управления АСОДУ;

управления технологическими процессами АСУТП.

В дальнейшем предполагается интеграция системы КАПРИ с АСУ институтом и рядом АСУ экспериментом и обработкой результатов научных исследований.

Основные функции указанных подсистем интегрированной машиностроительной системы КАПРИ:

САПР-К — автоматизированное конструкторское моделирование, компоновка, конструирование типовых сборочных узлов, типовых и оригинальных деталей, включая информационное обеспечение процесса конструирования, проведение конструкторских расчетов, подготовку текстовой документации и, при необходимости, получение чертежей;

САПР-Т — автоматизированное проектирование технологии обработки типовых и оригинальных деталей, включая информационное обеспечение, выбор заготовок, формирование маршрутной и операционной технологии, выбор технологических баз, расчет режимов обработки, выбор или конструирование режущего, измерительного инструмента и оснастки, нормирование трудоемкости производства, построение операционных видов полуфабриката, подготовка текстовой документации, генерация управляющих программ для станков с ЧПУ и промышленных роботов, группирование маршрутной и операционной технологии, имитационный контроль качества управляющих программ;

АСОЭУ — обработка портфеля заказов, технико-экономическое и оперативное планирование конструкторских работ и производства, управление материально-техническим обеспечением, учет и анализ эффективности конструкторских работ и производства;

АСОДУ — формирование и текущая корректировка сменных планов производства для каждого рабочего места, координация и синхронизация работы оборудования (станков с ЧПУ, роботов, транспорта и складов) в соответствии со сменными планами, текущий учет хода производства и состояния оборудования, слежение за движением изделий по техноло-

гическому маршруту, формирование отчетных данных и справок производственному персоналу о текущем состоянии производства, управление транспортно-складским комплексом;

АСУТП — автоматическая передача управляющих программ на станки с ЧПУ и роботы в соответствии со сменными планами, контроль технологических параметров, активный контроль размеров изделия в процессе обработки, контроль состояния оборудования, управление транспортными и складскими механизмами, испытание узлов.

ИМС КАПРИ создается на базе отечественных технических средств, серийно выпускаемых промышленностью. В настоящее время в эксплуатации или демонстрационном режиме используются версии подсистем ИМС КАПРИ, обеспечивающие конструирование некоторых типовых сборочных узлов, широкой номенклатуры осесимметричных деталей и их производство на программно-управляемом технологическом оборудовании под организационно-экономическим, оперативно-диспетчерским и технологическим автоматизированным управлением. При этом реализованы следующие функции системы КАПРИ:

САПР-К — конструирование типового расходомерного узла, осесимметричных типовых и оригинальных деталей с моделированием процесса обработки и визуализацией результатов моделирования, подготовкой текстовой и чертежной документации на машинных носителях, а при необходимости, в виде твердых копий;

САПР-Т — проектирование маршрутных и операционных карт изготовления осесимметричных деталей, генерация управляющих программ с контролем их качества посредством моделирования обработки, нормирование трудоемкости обработки;

АСОЭУ — планирование и учет производства, контроль материальной подготовки производства, анализ производства;

АСОДУ — формирование и учет выполнения сменных заданий, текущий учет и анализ выработки рабочих, прохождения заказов, простоев оборудования;

АСУТП — передача управляющих программ на станки с ЧПУ, коррекция управляющих программ, текущий контроль состояния станков.

Некоторые программные и аппаратные компоненты перечисленных подсистем подробнее описаны ниже.

Пакет программ «Синтез» используется для конструирования осесимметричных деталей. Конструктор получает на экране символьно-графического дисплея изображение детали, как бы нанизывая на ось геометрические элементы: цилиндр, конус, сферу, тор.

Работая в диалоге с ЭВМ, конструктор может задать технологические требования и получить автоматически подготовленные маршрутную и операционную карты и управляющую программу для изготовления детали на станке. При желании конструктор может получить на экране дисплея графическую интерпретацию технологического процесса изготовления детали в виде операционных эскизов и траектории режущего инструмента. Такой сквозной автоматизированный цикл конструирования и проектирования технологии удобен для выполнения экспресс-заказов.

Пакет программ СТЕП-Т позволяет существенно ускорить процесс подготовки технологии и программ для токарных станков с ЧПУ. Этот пакет автоматически рассчитывает сложные сопряжения поверхностей, выбирает заготовку, составляет управляющую программу и обеспечивает визуальный контроль качества на символьно-графическом дисплее.

Основные трудности проектирования технологических процессов для станков с ЧПУ в опытных производствах обусловлены тем, что технологи таких производств, как правило, не имеют специальной подготовки для работы с вычислительной техникой. Вместе с тем, при проектировании управляющих программ приходится решать такие задачи, как пересчет размеров в координаты опорных точек, разделение проходов на ходы и шаги, расчет координат опорных точек траектории, преобразование систем координат и другие, решение которых технологу несвойственно, весьма утомительно, и поэтому именно здесь делается основное число ошибок. Наиболее распространенные в настоящее время языки проектирования технологических процессов являются проблемно-ориентированными языками. Они дают только средства для решения задач (операторы расчета геометрических элементов, циклы механообработки, диагностика), но их использование не освобождает технолога от подробного описания траектории движения инструмента и команд управления технологическим оборудованием.

Задание процесса обработки деталей класса «тела вращения» при использовании пакета СТЕП-Т осуществляется в обычных для технолога терминах. Это достигнуто разработкой проблемно-ориентированного непроцедурного языка, позволяющего описать технологическую базу изготовления детали, обрабатываемые поверхности для каждого станова и последовательность обработки. При описании обрабатываемых поверхностей не нужно производить дополнительных расчетов — достаточно указать только размеры, проставленные на чертеже. Все остальные за-

дачи проектирования управляющих программ решаются автоматически.

В тех случаях, когда управляющая программа не может быть получена посредством пакета STEP-T, ее набирают на клавиатуре и редактируют на устройстве, именуемом станцией группового диспетчерского управления. Она состоит из микроЭВМ MC 1201.1, алфавитно-цифрового дисплея 15ИЭ-00-013, линий связи со станками, адаптеров связи «Телекварт» и активных адаптеров «Телепульт». Каждая станция группового диспетчерского управления может управлять шестью станками.

Программное обеспечение станции позволяет выполнять функции проектирования и редактирования управляющих программ, записи и чтения программ из банка программ, передачи программ на станки, чтения программ из процессора станка, опрос и запись состояния станка и данных об обработке управляющих программ за смену.

Подсистема организационно-экономического управления ежедневно формирует оперативный план производства в виде перечня деталей, назначенных к обработке в текущие сутки, сортированных по видам технологической обработки. При планировании используются данные учета и анализа состояния и простоев оборудования. Оперативный план передается в подсистему оперативно-диспетчерского управления. Работник плано-экономической группы в диалоге с ЭВМ превращает оперативный план производства в совокупность сменных парядов на каждый станок, оптимизированных по критерию минимума переналадок оборудования.

ИМС КАПРИ реализована на комплексе средств вычислительной техники, в состав которого входят ЭВМ ЕС-1045, четыре АРМ-М на базе ЭВМ СМ-4 (два на конструкторском и два на технологическом уровне), шесть автоматизированных рабочих мест конструкторов на базе символьно-графических станций «Графит», более сорока автоматизированных рабочих мест технологов, нормировщиков, начальников участков, мастеров и другого производственного персонала на базе алфавитно-цифровых дисплеев 15ИЭ-00-013.

ЭВМ и терминалы включены как абоненты в локальную сеть связи посредством двухпроводных телефонных линий. Станции связи выполнены на базе микроЭВМ MC 1201.1 и могут быть установлены в условиях цеха или КБ. Абоненты связываются с процессором микроЭВМ через адаптеры «Телекварт». Эта аппаратура позволяет одной станции обслуживать одновременно до 24 абонентов, удаленных от станции на расстояние до 7000 м. При помощи этих же блоков станции связи соединяются между собой. Программное обеспече-

ние станции допускает работу сети в неполной конфигурации и автоматическое восстановление работоспособности при сбоях или одновременном запуске станций. Предельная скорость передачи информации — 9600 бод. Изготовление деталей выполняется на токарных станках 16К20Т1, фрезерных станках ОФ 120 и обрабатывающем центре ИР-320 с микропроцессорными системами ЧПУ.

Создан и успешно работает роботизированный комплекс, состоящий из токарного станка 16К20Т1 и промышленного робота РК-1, разработанного совместно ИПМ им. М. В. Келдыша и ИАЭ им. И. В. Курчатова*. Система управления робота (на базе микроЭВМ «Электроника 60») допускает управление по семи независимым координатам и обеспечивает возможность простой настройки на новые операции, выполняемой с пульта дисплея и выносного пульта обучения робота.

Закончена разработка аппаратной части персональной графической станции «Пегас», предназначенной для графической интерпретации управляющих программ и графического отображения информации в подсистемах организационно-экономического и оперативно-диспетчерского управления. Станция выполнена в конструктиве алфавитно-цифрового дисплея 15ИЭ-00-013. Она работает с черно-белым изображением на растре 400×275 точек. Достоинство станции — наличие встроенной микроЭВМ MC 1201.1. Станция допускает подключение блоков для связи с сетями ЭВМ, графического манипулятора, накопителей на гибких магнитных дисках, устройства печати, а также стандартной аппаратуры микроЭВМ «Электроника 60».

Внедрение системы КАПРИ в полном объеме даст возможность повысить производительность труда при конструировании в 1,5...3 раза, технологической подготовке производства — в 2...4 раза, в производстве — в 1,5...3 раза и тем самым сократить сроки создания крупных экспериментальных стендов в 1,5...2 раза. Снизится потребность в работниках дефицитных профессий: конструкторах, технологах, рабочих-станочниках; улучшится структура выполняемых персоналом работ в пользу творческого труда.

Система КАПРИ — прообраз будущих автоматизированных комплексов инженерного обеспечения научных исследований.

Статья поступила 17 сентября 1985 г.

* Роботизированный механообрабатывающий комплекс для участка ГАП/Е. О. Адамов, С. И. Гримайло, С. С. Камынин и др. — см. следующую статью.

Е. О. Адамов, С. И. Гримайло, С. С. Камынин, Д. А. Корягин, Ю. П. Смольянов

РОБОТИЗИРОВАННЫЙ МЕХАНООБРАБАТЫВАЮЩИЙ КОМПЛЕКС ДЛЯ УЧАСТКА ГИБКОГО ПРОИЗВОДСТВА

Стремление повысить эффективность технологического оборудования и роботов, их универсальность и гибкость в переналадке, а также желание расширить возможности реагирования на различные производственные ситуации обусловили необходимость применения систем управления, базирующихся на микроЭВМ. В статье описываются аппаратно-программные средства роботизированного комплекса, управляемого от микроЭВМ и созданного Институтом атомной энергии им. И. В. Курчатова и Институтом прикладной математики им. М. В. Келдыша. На рассматриваемой стадии разработки комплекс позволяет автоматически изготавливать детали типа тел вращения.

Состав и структура связи аппаратных средств комплекса

Основу аппаратных средств комплекса составляет токарный станок 16К20Т1 с микропроцессорной системой управления «Электроника НЦ-31» и робот РК-1 (рис. 1). В состав комплекса входят также загрузочный стол, на котором устанавливается переналаживаемая кассета для заготовок и деталей, и измерительная позиция (рис. 2).

Робот состоит из универсального электромеханического манипулятора, построенного по модульному принципу, и системы управления. Манипулятор работает в цилиндрической системе координат, имеет семь степеней свободы — три транспортные, две ориентирующие и по одной степени в каждом из двух схватов. Приводится в движение электродвигателями постоянного тока типа МУ-50 (напряжение питания — 27 В, мощность — 75 Вт). Положение звеньев определяется с помощью фотоимпульсных датчиков ВЕ-178. Передаточные отношения редукторов датчиков подобраны таким образом, что одному импульсу с датчика соответствует перемещение на величину, не превышающую 0,1 мм. Точность позиционирования манипулятора определяется величиной порядка 0,3 мм. Грузоподъемность манипулятора 20 кг — при номинальной скорости движения (0,5 м/с — для линейных степеней свободы

и 80 град/с — для поворотных) и 30..40 кг — при пониженной. Масса — около 200 кг. Система управления робота выполнена по принципу полностью цифрового управления, что значительно упрощает ее, уменьшает габаритные размеры при невысокой стоимости, существенно повышает надежность работы.

В качестве управляющей ЭВМ выбрана микроЭВМ «Электроника 60», вычислительные и логические возможности которой позволяют реализовать весьма простые алгоритмы программирования действий робота, организовать поведение робота в условиях распознавания внешних ситуаций на основании показаний датчиков, выбора того или иного варианта выполнения двигательной задачи в зависимости от сложившейся внешней ситуации и т. п. Для управления манипулятором датчики обратной связи, определяющие положение звеньев, передают информацию в вычислительную машину, а все управляющие воздействия рассчитываются по программе. Это увеличивает нагрузку на управляющую ЭВМ, но обеспечивает большую гибкость в управлении.

Связь управляющей ЭВМ с манипулятором осуществляется посредством исполнительной системы, построенной также по модульному принципу. Исполнительная система вместе с источником питания и управляющей микроЭВМ «Электроника 60» монтируется в стойку управления. Число модулей исполнительной системы, одновременно устанавливаемых в стойке, ограничивает число управляемых степеней свободы до восьми. Каждой степени свободы соответствуют два модуля. Один осуществляет подсчет числа импульсов с датчика положения и преобразует его в цифровой код, пропорциональный относительному положению соответствующей степени свободы. Другой преобразует значения управляющих воздействий, рассчитанных на ЭВМ, в широтно-импульсный сигнал напряжения и выдает его на электродвигатель соответствующей степени свободы. Управляющими параметрами являются относительная продолжительность импульсов и их знак. Относительная продолжительность импульсов может принимать 64 различных значения: 0,1/63, 2/63, ..., 63/63. С учетом знака, определяемого «реверсом», воз-

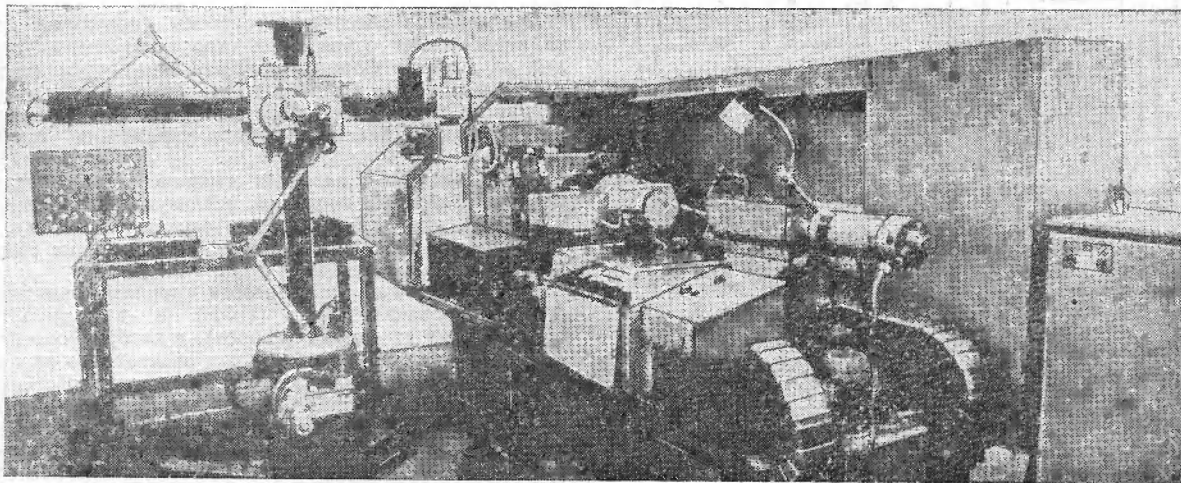


Рис. 1. Общий вид роботизированного комплекса

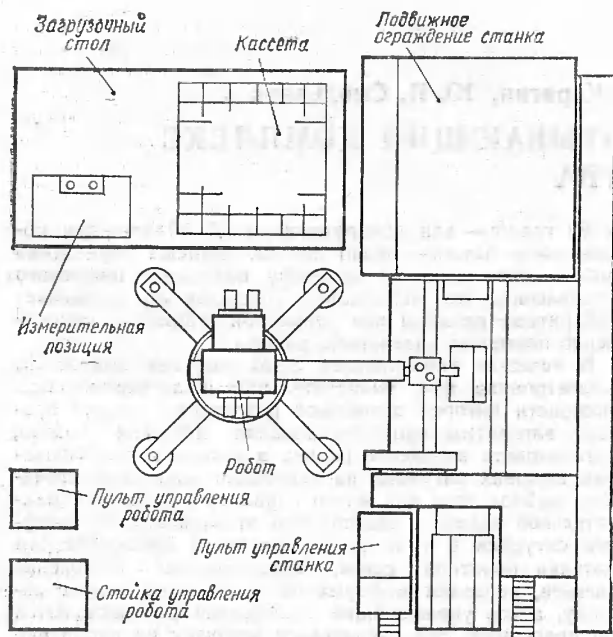


Рис. 2. Схема роботизированного комплекса (вид сверху)

возможные значения управления будут $\pm 1/63$, $\pm 2/63$, ..., $\pm 63/63$. Частота следования импульсов постоянна и равна 400 Гц.

Для организации группового управления несколькими манипуляторами от одной ЭВМ к стойке управления, содержащей управляющую микроЭВМ (эта стойка называется ведущей), подключается одна или несколько стоек без микроЭВМ (эти стойки содержат только исполнительную систему с источниками питания и называются ведомыми). В стойке управления предусмотрен коммутатор, позволяющий оператору вручную под-

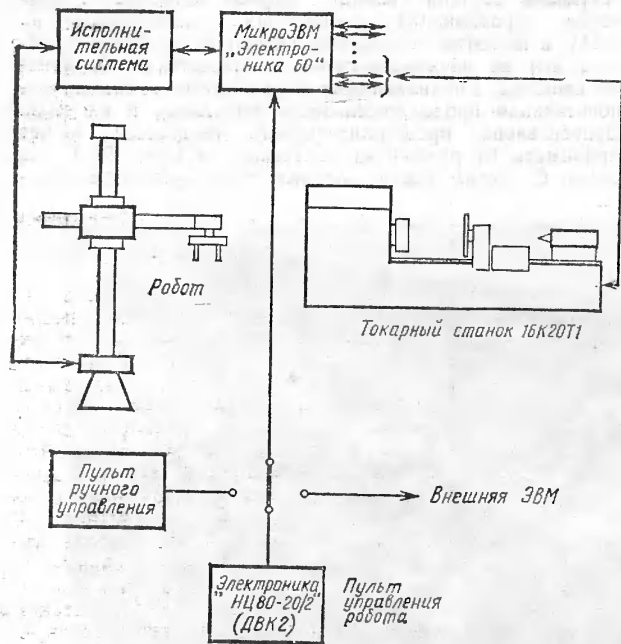


Рис. 3. Структурная схема связи аппаратных средств роботизированного комплекса

ключить управляющую ЭВМ к пульту управления, пульту ручного управления, внешней ЭВМ, с которых осуществляется загрузка программ и данных в память микроЭВМ, а также запуск программы. Управляющая ЭВМ имеет 16 входных и 16 выходных бинарных каналов связи с внешним оборудованием. Таким образом, стойка управления робота совместно с манипулятором представляет собой отдельный модуль, обеспечивающий простые и гибкие связи с различными устройствами. Путем подключения станка к стойке управления может быть получен модуль Станок-робот.

Выносной пульт ручного управления используется для программирования робота и ручного управления. Пульт управления на основе диалогового вычислительного комплекса «Электроника ИЦ-80-20/2» предназначен для организации архива программ на ГМД, программирования робота и обмена сообщениями между оператором и системой управления (рис. 3). ДВК-2 одновременно играет роль центральной ЭВМ робототехнического участка, к которой посредством разрабатываемого коммутатора может быть подключено до 10 модулей Станок-робот. В этом случае ДВК-2 определяет задачи, решаемые каждым из модулей, и контролирует их выполнение.

Управление станком осуществляется через блок связи, который позволяет имитировать нажатие основных клавиш пульта управления УЧПУ станка по сигналу от ЭВМ, получать информацию о срабатывании некоторых из них, а также управлять сжатием-разжатием кулачков зажимного патрона станка и подводом-отводом его задней бабки. Разработанные алгоритмы обеспечивают автоматический запуск на выполнение любой из управляющих программ, находящихся в памяти УЧПУ станка.

Блок связи не позволяет однако осуществлять «подкачку» управляющих программ с внешнего носителя, полный контроль состояния УЧПУ и т. п. Эти возможности автоматического управления станком можно реализовать с помощью интерфейса, получившего название информационно-пультового канала (ИПК), разработка которого завершается в настоящее время. Благодаря ИПК станок так же, как и робот, становится отдельным модулем, обеспечивающим возможность подключения к любому управляющему устройству (например, к стойке управления робота, центральной ЭВМ робототехнического участка, внешней ЭВМ). Устройство с ИПК получает все возможности управления станком, которые имеет оператор-станочник.

Программное обеспечение системы управления робота

Программное обеспечение системы управления робота представляет собой множество программных модулей, из которых формируются различные управляющие программы (УП). Среди них основные: УП-1, рассчитанная на управление одним роботом, и УП-2, которую предполагается реализовать для управления группой роботов от одной микроЭВМ.

Программа УП-1 включает диспетчер, интерпретатор, набор стандартных программ, реализующих выполнение различных операций, цифровую следящую систему.

Диспетчер выполняет функции, связанные с организацией работы интерпретатора.

Интерпретатор способен принять программу функционирования робота (ПФР) и выполнить ее. ПФР состоит из команд (кадров), в которых определяются код выполняемой операции, значения ее параметров и указываются команды, на одну из которых должно быть передано управление вслед за выполнением данной. По своей структуре ПФР имеет вид ориентированного графа. Выбор команды зависит от результата выполнения предыдущей команды, что позволяет организовать адаптивное поведение робота.

Таблица 1

Объем памяти, занимаемой основными модулями управляющей программы УП-1

Модуль	Объем памяти, Кбайт
Диспетчер	
алгоритм диспетчера (совместно со служебными подпрограммами)	3,1
рабочие массивы и данные	0,4
Интерпретатор	
алгоритм интерпретатора (совместно со служебными подпрограммами)	4,4
рабочие массивы	0,5
массивы для хранения команд ПФР	31,3
Стандартные программы, реализующие выполнение различных операций:	
управление действиями робота	3,0
управление порядком выполнения ПФР	2,0
формирование команд	1,5
Следящая система	
алгоритм следящей системы	0,6
параметры, данные и рабочие массивы	0,6
Общий объем управляющей программы УП-1	47,4

Набор стандартных программ, реализующих выполнение различных операций, определяет «интеллектуальные» («профессиональные») способности робота. Этот набор является открытым множеством (при желании в него могут быть включены дополнительные программы, а также организована замена или исключение программ) *.

Цифровая следящая система осуществляет непосредственное управление степенями свободы робота, обрабатывая нужные программные движения. Описание программных движений поступают в цифровую следящую систему из стандартных программ, реализующих выполнение различных операций. В настоящее время используется следящая система позиционного типа, обеспечивающая возможность одновременного перемещения степеней свободы в заданные положения с заданными скоростями. Время, приходящееся на вычисление управляющего воздействия для одной степени свободы, составляет приблизительно 1...1,5 мс. Распределение памяти между модулями управляющей программы показано в табл. 1.

Работа программы УП-1 предполагает наличие двух взаимодействующих процессов. В одном (фоновом) — работают диспетчер, интерпретатор и стандартные программы, в другом (основном) — следящая система, управление которой передается при наступлении каждого прерывания от таймера (частота прерываний для ЭВМ «Электроника 60» постоянна и равна 50 Гц). Остаток времени отдается фоновому процессу.

Для управления роботом разработан язык программирования, позволяющий вводить, редактировать ПФР

* Операции ПФР могут определять функционально законченные сложные действия робота, поэтому язык задания ПФР может иметь высокий уровень.

и запускать их на исполнение. Использование языка позволяет оператору работать с системой управления в интерактивном режиме. Уровень языка зависит от набора стандартных программ, реализующих различные операции. Для организации гибкого и многопланового управления роботом в программу УП-1 введены 16 операций управления действиями робота, 27 операций управления порядком выполнения ПФР и 8 операций формирования команд (ОФК). ОФК позволяют автоматизировать задание ПФР. Каждая ОФК осуществляет запрос значений параметров с помощью развитых диалоговых средств, ориентированных на конкретную проблему, а затем автоматически формирует команду обращения к соответствующей операции с указанными значениями параметров. ОФК не имеют параметров, поэтому обращение к ним является предельно простым.

Программа УП-2 является простейшим обобщением программы УП-1 на случай управления группой роботов. Фоновый процесс в данном случае разбит на несколько (по числу роботов) подпроцессов. Каждый подпроцесс включает интерпретатор, принимающий и выполняющий программу функционирования соответствующего робота и имеющий те же возможности и тот же язык задания ПФР, что и интерпретатор программ УП-1. Каждый подпроцесс имеет свой набор стандартных программ, реализующих выполнение различных операций (различные подпроцессы могут иметь идентичные стандартные программы). Однако стандартные программы, занимающие большой объем памяти ЭВМ и реализующие выполнение относительно редко используемых операций, имеются в УП-2 в единственном экземпляре.

Набор операций для каждого робота при решении одних и тех же задач, в основном, совпадает с набором операций управляющей программы УП-1. Однако для того, чтобы скоординировать работу нескольких роботов, управляемых от одной ЭВМ, в этот набор необходимо добавить операции обмена управляющими сигналами между различными ПФР: Чтение сигнала, Ожидание сигнала, Посылка сигнала и т. п. За распределение времени между подпроцессами программы отвечает диспетчер. Он предоставляет также различным подпроцессам возможность использования программ и устройств, имеющихся в системе управления в единственном экземпляре.

Цифровая следящая система, управляемая степенями свободы всех роботов, подключенных к микроЭВМ «Электроника 60», отличается от цифровой следящей системы программы УП-1 только увеличенным числом обслуживаемых степеней свободы.

Программное обеспечение системы управления робототехнического участка

Существующее программное обеспечение системы управления дает возможность формировать управляющие программы, способные обслужить один модуль Станко-робот, подключенный к центральной ЭВМ. В функции обслуживания входят обмен управляющими программами и программами функционирования робота между памятью микроЭВМ и архивом на ГМД, запуск работы управляющей программы системы управления робота (после этого ДВК начинает выполнять функции организации обмена сообщениями между оператором и системой управления робота), выполнение действий по проверке работоспособности робота и его системы управления, а также организация работы с файлами на диске. Пока отсутствует возможность хранения в архиве на ГМД управляющих программ УЧПУ станка, их загрузка в память УЧПУ осуществляется вручную, с пульта управления станка. Работа с системой происходит в интерактивном режиме.

Ближайшие перспективы развития программного обеспечения системы управления робототехнического участ-

ка связаны с разработкой информационно-пультового канала, позволяющего подключать станок к центральной ЭВМ и управляющей ЭВМ робота и одновременно управлять десятью модулями. В данном случае управляющая программа осуществляет опрос состояний модулей, интерпретирует программы функционирования модулей, а также позволяет оператору обмениваться сообщениями с каждым из них. Состав и структура управляющей программы аналогичны составу и структуре управляющей программы УП-2. Используемый набор операций содержит операции чтения ПФР из архива на ГМД, загрузки и микроЭВМ, запуска на исполнение,

загрузки управляющих программ в память УЧПУ станка и т. п.

Совокупность одновременно выполняемых в ДВК программ функционирования модулей определяет программу функционирования робототехнического участка. Эта программа может быть задана вручную с клавиатуры дисплея. Для повышения эффективности программирования и обеспечения простоты и удобства в разработке управляющих программ УЧПУ станка предполагается использовать систему автоматизации программирования, включающую следующие две основные части: систему технологической подготовки данных для то-

Таблица 2

Основные операции обслуживания токарного станка

Номер операции	Выполняемая функция	Параметры операции	Основные контрольные проверки и элементы адаптации
1	Захват заготовки из кассеты с измерением ее длины и диаметра	Номер места в кассете (по нему определяются соответствующие координаты робота) Минимально допустимый диаметр заготовки Минимально допустимая длина заготовки Высота захвата заготовки	Контроль наличия заготовки в кассете Проверка соответствия длины заготовки допустимому диапазону Проверка соответствия диаметра заготовки допустимому диапазону
2	Установка готовой детали в кассету	Номер места в кассете	Контроль правильности установки детали в кассету
3	Установка заготовки в зажимной патрон	Координаты робота при установке заготовки в патрон	Контроль «утапливания» заготовки в зажимном патроне Контроль «досылки» заготовки в патрон Проверка наличия заготовки в патроне после сжатия кулачков Контроль зажатия заготовки в патроне
4	Вынимание детали из зажимного патрона	Координаты робота при вынимании детали из патрона	Контроль наличия детали в патроне перед началом вынимания
5	Контроль правильности изготовления детали по внешнему диаметру с целью проверки износа или поломки инструмента и последующий захват детали	Координаты робота при измерении диаметра детали Высота для измерения диаметра детали	Контроль соответствия диаметра детали допустимому диапазону
6	Контроль правильности изготовления детали по внутреннему диаметру с целью проверки износа или поломки инструмента и последующий захват детали	Координаты робота при попытке насаживания детали на непроходной калибр Координаты робота при попытке насаживания детали на проходной калибр Высота установки детали на непроходной калибр Высота установки детали на проходной калибр Высота захвата детали после контроля внутреннего диаметра	Проверка «наживления» детали на непроходной калибр Проверка «насаживания» детали на непроходной калибр Проверка «наживления» детали на проходной калибр Проверка «насаживания» детали на проходной калибр Проверка наличия детали в схвате при попытке ее захвата после контроля внутреннего диаметра
7	Открывание подвижного ограждения станка	—	Контроль правильности открывания путем проверки наличия ручки ограждения в схвате робота до и после открывания
8	Закрывание подвижного ограждения станка	—	Контроль правильности закрывания путем проверки наличия ручки ограждения в схвате робота до и после закрывания
9	Ожидание роботом окончания выполнения управляющей программы УЧПУ	Время ожидания	Проверка готовности станка к работе Проверка истечения заданного времени ожидания
10	Запуск выполнения управляющей программы изготовления детали в УЧПУ станка	Измеренная длина заготовки Номинальная длина заготовки Измеренный диаметр заготовки Номинальный диаметр заготовки Номер управляющей программы УЧПУ станка (по номеру автоматически определяется адрес запуска ее выполнения)	Проверка готовности станка к работе Проверка запуска работы управляющей программы с указанным номером Корректировка начальной точки обработки заготовки в зависимости от измеренных роботом ее длины и диаметра

карного станка (СТЕП-Т) и программу-планировщик. Первая позволяет с помощью языка высокого уровня вводить в память ДВК описание геометрии детали в виде последовательности стандартизованных элементов с указанием соответствующих размеров детали. Планировщик предоставляет оператору возможность в интерактивном режиме с помощью языка высокого уровня указать тип и число изготавливаемых деталей (из числа специфицированных с помощью программы СТЕП-Т). Далее автоматически формируется программа функционирования робототехнического участка.

Каждая программа, формируемая в рамках программного обеспечения системы управления робототехнического участка, представляет собой задачу, которая запускается на выполнение из монитора. После окончания выполнения задачи осуществляется выход в монитор и запуск новой задачи. Имеется возможность прерывания выполнения задачи с последующим выходом в монитор.

Операции, выполняемые роботом при обслуживании станка

Операции по обслуживанию роботом токарного станка с УЧПУ имеют высокий уровень и реализованы на языке задания ПФР (табл. 2). Во все операции введены контрольные проверки, а в некоторые из них — элементы адаптации. В случае обнаружения ошибочной ситуации выполнение ПФР останавливается, и на экран дисплея выдается диагностическое сообщение. В дальнейшем предполагается развить алгоритмы операций, с тем чтобы робот мог предпринимать конкретные действия по устранению возникающих ошибок.

Характерной особенностью алгоритмов рассматриваемых операций является то, что различные измерения и проверки в них осуществляются только на основе показаний датчиков положения степеней свободы манипулятора. Так, диаметр заготовки измеряется путем зажатия ее в захвате с последующим считыванием показания соответствующего датчика положения. Измерение длины заготовки осуществляется с помощью показания датчика положения механизма вертикального перемещения руки при выводе руки в упор и т. п.

Помимо основных контрольных проверок в операциях имеются также блокировки, останавливающие выполнение ПФР при обнаружении ошибок в логике выполнения программы (например, устанавливаемая в зажимной патрон станка заготовка должна быть предварительно зажата в захвате робота, а кулачки патрона при этом должны быть отжаты и т. д.). Такого рода блокировки осуществляются путем установки и проверки различными операциями признаков, описывающих состояние схватов робота и среды, в которой он работает.

Пример использования комплекса

При изготовлении деталей типа ниппель, требующих двусторонней обработки, необходимо дважды устанавливать их в зажимной патрон станка. После обработки одной стороны осуществляется контроль правильности изготовления по внешнему и внутреннему диаметрам (в это время станок обрабатывает другую деталь), а затем снова (но другой стороной) деталь устанавливается в зажимной патрон станка (рис. 4, 5). Готовые детали устанавливаются в кассету: одна половина кассеты используется для заготовок (рис. 6), другая половина — для деталей.

Обращение к операциям по обслуживанию роботом станка организуется в следующем порядке:

- ожидание роботом окончания выполнения управляющей программы УЧПУ станка (операция 9 в табл. 2);
- открывание подвижного ограждения станка (операция 7);

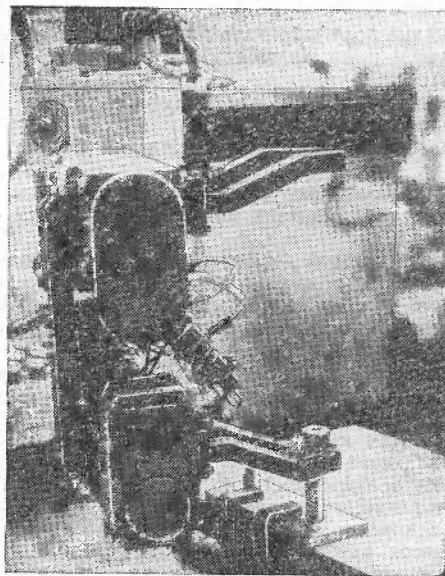


Рис. 4. Контроль правильности изготовления полуфабриката по его внутреннему диаметру

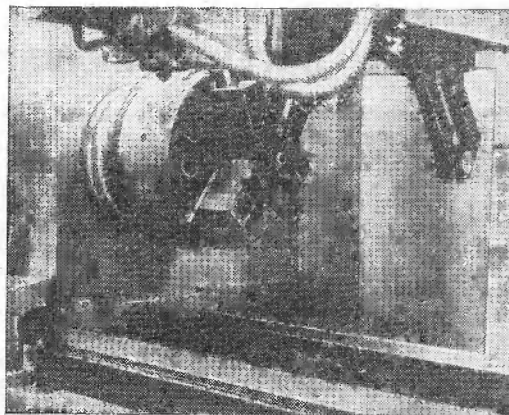


Рис. 5. Выпимание полуфабриката, полученного на предыдущем шаге, из зажимного патрона станка

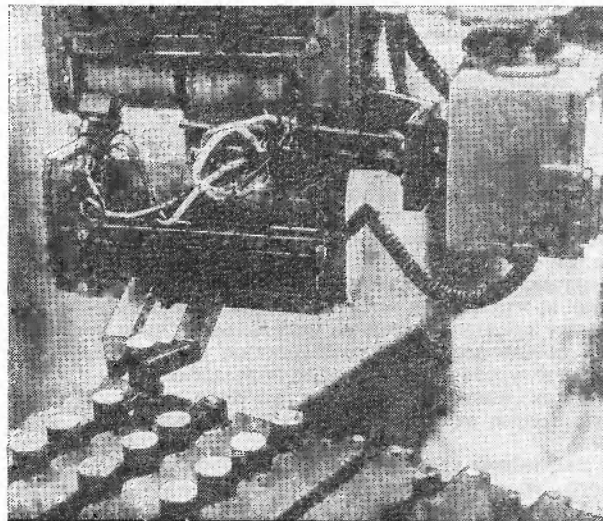


Рис. 6. Захват заготовки из кассеты

внимание полуфабриката или детали, полученных на предыдущем шаге, из зажимного патрона станка (операция 4);

установка заготовки или полуфабриката в зажимной патрон станка (операция 3);

закрывание подвижного ограждения станка (операция 8);

запуск выполнения управляющей программы УЧПУ станка для изготовления полуфабриката или детали (операция 10);

установка готовой детали в кассету (операция 2), захват заготовки с измерением ее длины и диаметра (операция 1), если на предыдущем шаге была изготовлена деталь, либо контроль правильности изготовления полуфабриката (операции 5 и 6).

Управляющая программа УЧПУ станка состоит из двух основных частей, соответствующих обработке первой и второй сторон изготавливаемой детали. Помимо этого, имеется несколько дополнительных программ, осуществляющих настройку начальной точки черновой обработки в зависимости от измеренных роботом диаметра и длины заготовки. Каждой из указанных программ присвоен номер, по которому автоматически определяется адрес запуска ее на выполнение по сигналу от системы управления робота.

УДК 681.32

А. И. Савин, М. Н. Кузнецов, А. С. Уксусов, А. Н. Целяпин

СТРУКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ГАП

Раздельное применение автоматизированного оборудования с интеллектуальными системами управления, станков с ЧПУ, промышленной робототехники, автоматических складов, транспортных систем и систем управления на базе микро- и мини-ЭВМ не позволило полностью использовать их возможности. И только объединение их в единую производственную систему позволяет эффективно функционировать автоматизированному производству. Центральное место в гибких автоматизированных производствах (ГАП) занимают микропроцессорные средства управления [1].

Основная задача организации — это понимание производственного процесса и выбор для различных уровней этого процесса структуры подразделений, их функциональных взаимосвязей (форм организации), наиболее подходящих для рассматриваемого производства.

Традиционное производство выполняет четыре основные функции: составление производственной программы, подготовка производства, выполнение производственной программы, контроль и управление ходом производства.

Большинство реально существующих структур управления последовательно реализуют эти функции, сравнивая производственную программу с выполнением плана в конце декады, месяца, квартала.

При каких-либо возмущающих факторах решение принимает цеховое руководство, а чаще всего мастера участков, которые пока находятся практически в информационной изоляции, проявляющейся «по горизонтали» (отсутствие связи между подразделениями предприятия, выполняющими одну и ту же производственную программу) и «по вертикали» (отсутствие связи между производственными подразделениями и плановыми, производственно-диспетчерскими, руководством различного уровня, даже внутри подразделения).

Такая изоляция ухудшает качество информации, а это, в свою очередь, нарушает производственный процесс.

Важнейшие критерии качества информации — актуальность, полнота, точность.

Актуальность. Данный критерий качества, к сожалению, хуже всего реализуется на практике. Это

Ниже приведен перечень основных действий, характеризующих работу роботизированного комплекса при изготовлении деталей типа «шпindel» и среднее время их выполнения в минутах: несредственное обслуживание роботом станка (открытие подвижного ограждения, вынимание полуфабриката или детали из патрона и т. д.) — 2; установка готовой детали в кассету и захват заготовки — 1,3; контроль правильности изготовления полуфабриката по внешнему и внутреннему диаметрам — 2; обработка на станке первой стороны изготавливаемой детали — 12,5; обработка на станке второй стороны изготавливаемой детали — 3.

ЛИТЕРАТУРА

1. Гибкие производственные комплексы / Под ред. П. Н. Белянина, В. А. Лещенко. — М.: Машиностроение, 1984. — 383 с.

2. Белянин П. Н. Промышленные роботы и их применение. — М.: Машиностроение, 1983. — 310 с.

3. Промышленная робототехника / Под ред. Я. А. Шифрина. — М.: Машиностроение, 1982. — 415 с.

Статья поступила 30 августа 1985 г.

происходит из-за отсутствия прежде всего необходимых технических средств сбора и регистрации информации и несвоевременности поступления этой информации из-за недисциплинированности соответствующих исполнителей. Актуальность информации позволяет сделать ПРОГНОЗ дальнейшего развития производственной ситуации для принятия решения.

Полнота отражает желание обладать всей имеющейся для принятия решения информацией (объем и количество). Если получить полную информацию невозможно, прибегают к статистическим методам. Однако всюду, где это возможно, получение полной информации, точно отражающей все аспекты производственной ситуации, значительно ценнее для анализа и понимания ситуации, чем получение ее обработкой сложными методами, вычислением «вероятности» этого события.

Точность — дополняет два предыдущих критерия. Она гарантирует, что не произойдет никакого искажения, которое могло бы видоизменить один или несколько компонентов информации (это может привести к полному искажению суммарной информации).

Очевидно, что вся необходимая информация должна поступать своевременно именно на тот уровень организационной структуры, который должен принять решение и может его реализовать.

Основной технической компонент, обеспечивающий гибкость производства, — это системы управления на базе микропроцессорной техники, объединенные в единую интегрированную систему с помощью сети передачи данных. Отсюда следует вывод о тесной связи, единстве организационной структуры гибкого автоматизированного производства и структуры систем управления ГАП (рис. 1). На рис. 1 в верхней части приведены уровни организационной структуры производства, а в нижней — соответствующие им функциональные системы управления ГАП.

Техническая основа ГАП — это современное технологическое оборудование, управляемое микропроцессорной системой (МПСУ).

Гибкий производственный модуль (ГПМ) — это производственная ячейка, состоящая из единицы технологического оборудования, оснащенная МПСУ и средств

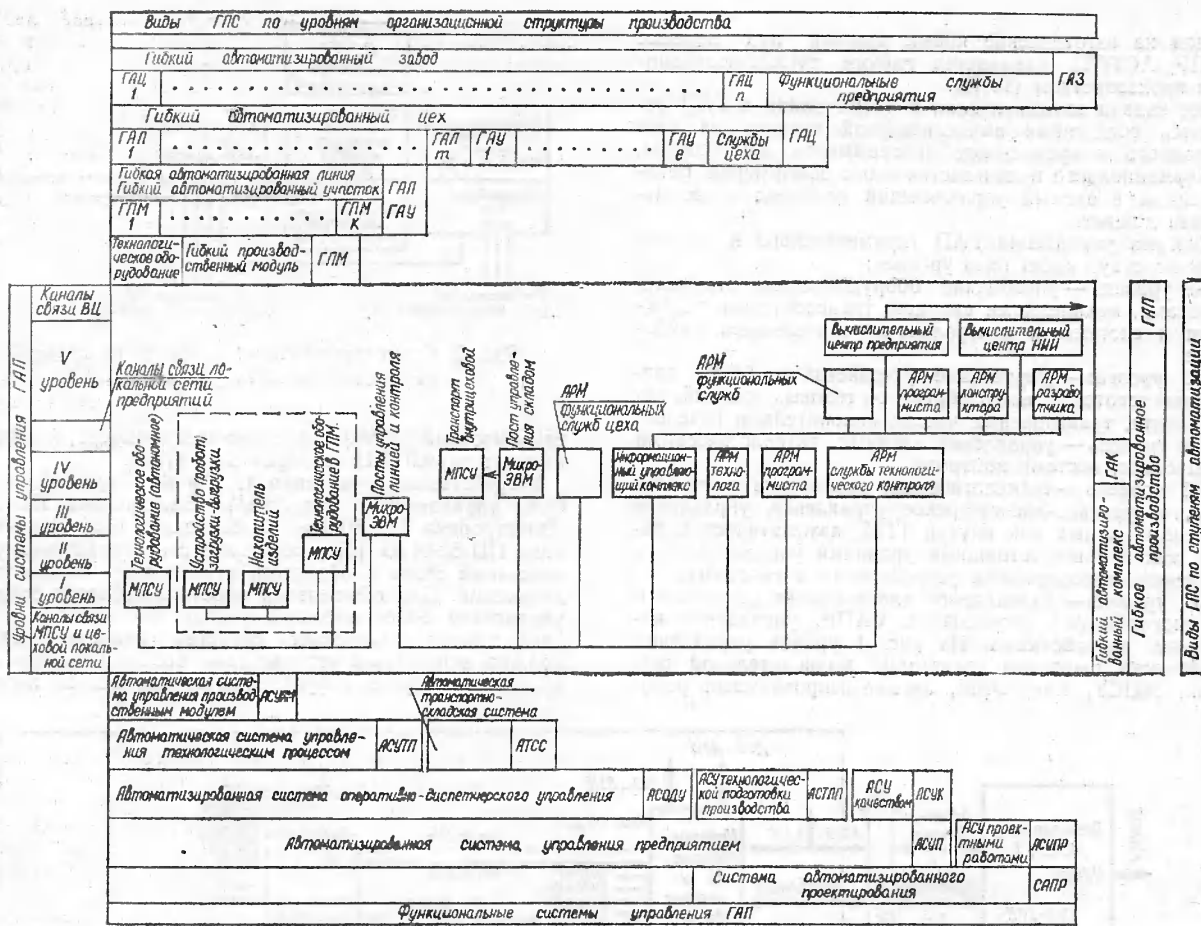


Рис. 1. Структурная схема ГПС и интегрированной системы управления ГАП

вами автоматизации технологического процесса, автономно функционирующая, осуществляющая многократные циклы и имеющая возможность встраивания в систему более высокого уровня [2].

Средства автоматизации могут включать в себя устройства загрузки-выгрузки, удаления отходов, запасы технологической оснастки, а также устройства, например для автоматического контроля и диагностики.

Гибкий производственный модуль — основная ячейка в организационной структуре гибких автоматизированных производств.

Несколько ГПМ, объединенных автоматизированной системой управления и технологическим маршрутом, называются гибкой автоматизированной линией (ГАЛ).

Если гибкая производственная система, состоящая из нескольких ГПМ, объединенных автоматизированной системой управления и технологическим маршрутом, предусматривает возможность изменения последовательности использования технологического оборудования, то она называется гибким автоматизированным участком (ГАУ).

Гибкий автоматизированный цех (ГЦ) — это ГПС, представляющая собой совокупность гибких автоматизированных линий и, в общем случае, гибких автоматизированных участков. Совокупность гибких автоматизированных цехов, предназначенная для выпуска готовых изделий в соответствии с планом производства, составляет гибкий автоматизированный завод (ГАЗ).

В состав интегрированной системы управления ГАП входят:

функциональные автоматические системы управления —

технологическим оборудованием АСУ ТО [3], гибким производственным модулем (АСУ ПМ), технологическим процессом (АСУ ТП), технологической подготовкой производства (АСТПП), качеством (АСУК), предприятием (АСУП), проектными работами (АСУПР); автоматическая транспортно-складская система (АТСС);

автоматизированная система оперативно-диспетчерского управления (АСОДУ);

система автоматизированного проектирования (САПР). По ступеням автоматизации ГПС подразделяется на гибкий автоматизированный комплекс (ГПК) и гибкое автоматизированное производство (ГАП).

Гибкий производственный комплекс — это гибкая производственная система, состоящая из нескольких гибких производственных модулей, объединенных автоматизированной системой управления и транспортно-складской системой, автономно функционирующая в течение заданного времени и имеющая возможность встраивания в систему более высокой степени автоматизации.

Важнейший компонент ГПК — автоматизированная транспортно-складская система, состоящая из транспортной и складской подсистем.

Совместное функционирование ГПМ и АТСС обеспечивается системой технологической подготовки производства и оперативно-диспетчерского управления.

Гибкая производственная система, состоящая из одного или нескольких ГПК, объединенных автоматизированной системой управления (АСУП, АСОДУ) и транспортно-складской автоматической системой, и перехо-

дящая на изготовление новых изделий при помощи САПР АСТПП, называется гибким автоматизированным производством (ГАП).

Все задачи автоматизации и оптимизации в ГАП решаются средствами вычислительной техники на базе системного и прикладного программного обеспечения, информационного и лингвистического обеспечения, объединенных в единый управляющий комплекс сетью передачи данных.

Система управления ГАП (применительно к любому производству) имеет пять уровней:

1-й уровень — управление оборудованием, станками, роботами, механизмами складов, транспортными тележками и системами, контрольно-измерительными приборами;

2-й уровень — управление устройствами ГПМ, движением заготовок, полуфабрикатов готовых изделий, инструмента, транспортных кассет, накопителями изделий;

3-й уровень — управление линиями, технологическими процессами, постами контроля;

4-й уровень — технологическая подготовка производства, оперативно-диспетчерское управление, управление обменом данных как внутри ГПК, автоматической линии, так и с вышестоящими уровнями управления, управление транспортными устройствами и складами;

5-й уровень — календарное планирование основного и вспомогательного производств, САПР, управление качеством, разработками. На рис. 1 уровни управления графически выделены средствами вычислительной техники: МПСУ, микроЭВМ, автоматизированными рабо-

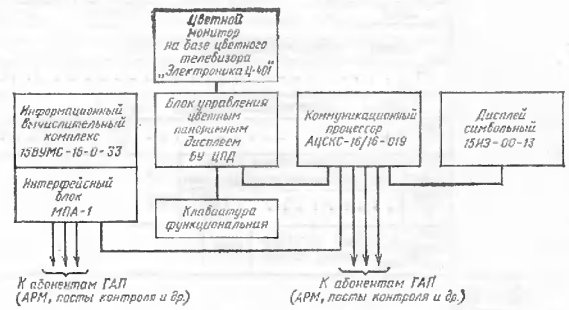


Рис. 2. Структурная схема технических средств центрального поста управления

чим местами (АРМ) технолога-программиста, функциональных служб, ВЦ предприятия и НИИ.

Устройствами управления 1, 2 и 3-го уровней могут быть управляющие микроЭВМ «Электроника НЦ-31», «Электроника НЦ-80-31», «Электроника 60», «Электроника НЦ-80-01Д» [4] в сочетании с необходимыми устройствами связи с объектом, усилителями мощности и датчиками. Для обеспечения работоспособности средств управления более высокого уровня или каналов передачи данных технические средства нижнего уровня должны оснащаться устройствами внешней памяти для хранения программы функционирования или на партию

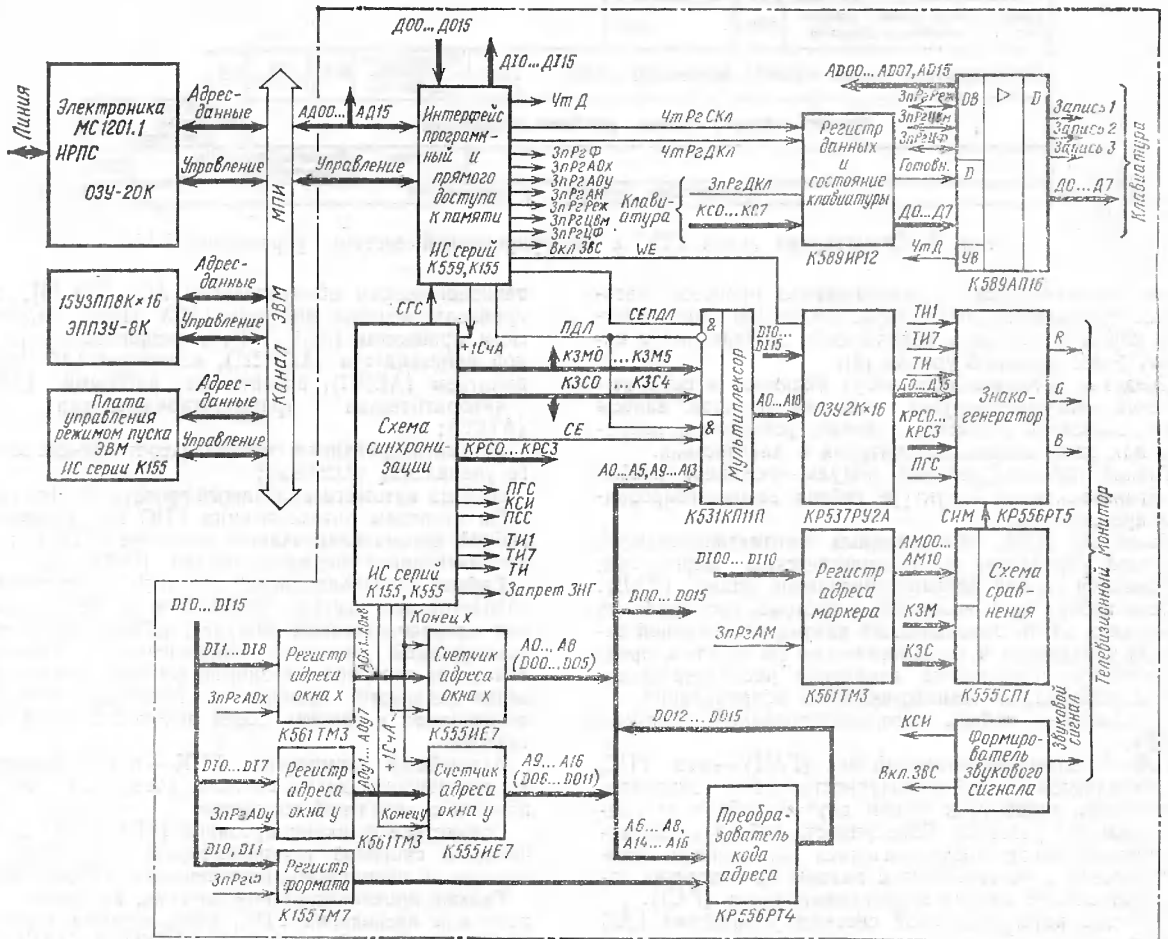


Рис. 3. Структурная схема блока управления цветного панорамного дисплея

обрабатываемых изделий, или на реализацию определенной технологической операции, например загрузки-выгрузки склада без необходимости обращения к ресурсам технических средств вышестоящего уровня управления.

Технические средства 4-го уровня состоят из центрального поста управления; автоматизированных рабочих мест работников функциональных служб, технолога-программиста, служб ОТК.

С помощью центрального поста управления решают задачи обработки, отображения, накопления, печати и передачи информации для оперативно-диспетчерского управления и связи с АРМ.

В состав центрального поста управления (рис. 2) входят информационно-вычислительный комплекс 15ВУМС-16—33 на базе мини-ЭВМ «Электроника 100-25» с интерфейсным блоком; коммуникационный процессор АЦКС; черно-белый символьный дисплей типа 15ИЭ-00—013; цветной панорамный дисплей (ЦПД); промышленная телевизионная установка ПТУ-61.

Коммуникационный процессор АЦКС передает информацию между процессорными устройствами 1, 2 и 3-го уровней управления и служит для связи трех уровней с четвертым.

В состав коммуникационного процессора входят: одноплатная микроЭВМ МС 1201.1; ППЗУ типа 15УЗПП-8К×16—9; ОЗУ емкостью 4К 16-разрядных слов; модули последовательного обмена и источник питания.

На экране цветного монитора или модифицированного цветного телевизионного приемника информация

отображается в виде мнемосхем станков, роботов, складов, роботизированных комплексов с выведением информации о состоянии оборудования, движении партий изделий и другой промышленной информации, поступающей в центральный пост управления в реальном масштабе времени. Отличительная функциональная особенность блока управления цветным панорамным дисплеем (БУ ЦПД — см. рис. 3) — это возможность организации режима «просмотрового окна» для информации, записанной на 8 экранах.

Технические средства центрального поста были изготовлены и успешно прошли эксплуатационные испытания.

За справками обращаться по тел. 534-08-53 (Москва).

ЛИТЕРАТУРА

1. Кузнецов М. Н., Савин А. И. Задачи внедрения систем управления оборудованием на базе микропроцессоров и микроЭВМ. — Электронная техника. Сер. 7. ТОПО, 1983, вып. 1(116), с. 3—4.

2. ГОСТ 26228—84. Системы производственные гибкие. Термины и определения.

3. Кузнецов М. Н., Мухин Б. С., Плешаков В. А. Функции микропроцессорных систем управления специальным технологическим оборудованием. — Электронная техника. Сер. 7. ТОПО, 1983, вып. 1(116), с. 9—11.

4. Толстых Б. Л. Развитие средств вычислительной техники. — Электронная промышленность, 1983, вып. 1, с. 37—39.

Статья поступила 6 августа 1985 г.

УДК 681.3.06—181.4

Ю. М. Соломенцев, В. Л. Сосонкин

ОПЕРАТИВНОЕ УПРАВЛЕНИЕ ГИБКИМИ ПРОИЗВОДСТВЕННЫМИ СИСТЕМАМИ

Функция гибкой производственной системы (ГПС) — изготовление заданных партий деталей. Назначение системы оперативного управления — наиболее эффективно использовать ресурсы ГПС для изготовления деталей, приоритет и запуск которых в производство обусловлены расписанием (оперативным планом).

В подобной формулировке назначение оперативного управления напоминает задачи операционной системы реального времени при управлении ресурсами вычислительной системы. Опишем работу системы оперативного управления ГПС в терминах операционных систем. Каждой обрабатываемой детали сопоставим понятие «задача для ГПС». Введенные задачи (детали) попадают в поле зрения системы оперативного управления, которая и руководит их решением (изготовлением) в период прохождения через технологический мультипроцессор (оборудование ГПС). Каждая задача (деталь) располагает определенным приоритетом. Один из принципов назначения приоритета — на основе отношения предшествования, установленного планом-расписанием. Для равноприоритетных задач (или задач, приоритет которых не задан) можно использовать заранее выбранную дисциплину обслуживания, например «первый пришел — первый обслужен», выбор в пользу задачи-детали, длительность операции для которой наибольшая и др.

Задачи претендуют на ресурсы: технологические и транспортные «процессоры» (станки и транспортные средства), а также множество приспособлений и инструментов. Связи между задачами проявляются в том, что одна задача может запустить другую, освободив для нее ресурс и передав ей соответствующее сообщение.

Связи устанавливаются через системные структуры данных — обменники, в которых хранятся списки задач, ожидающих сообщения, или списки сообщений, которые ожидают задачи.

Пример обменника с задачами: список деталей, находящихся в накопителе и ожидающих освобождения рабочей зоны станка. Пример обменника с сообщениями: список свободных станков, ожидающих поступления детали для обработки.

Как и в операционных системах, сообщение, поступившее в обменник с задачами (деталью), ставится в соответствие первой ожидающей задаче (детали) в списке, и задача удаляется из списка. Это означает, что задача готова к выполнению (деталь готова занять технологический или вспомогательный «процессор»). Точно так же задача, поступившая в обменник с сообщениями, ставится в соответствие первому ожидающему сообщению, и сообщение удаляется из списка. Это означает, что задача готова к выполнению (для детали приговорен технологический или вспомогательный «процессор»).

Задача (деталь) может находиться в одном из пяти состояний: задержанная (получившая требование условной или безусловной выдержки времени); ожидающая (находящаяся в обменнике); приостановленная (не располагающая необходимыми ресурсами); решаемая на «процессоре» (технологическом или вспомогательном); готовая (получившая ожидаемое сообщение, выполнившая требование условной или безусловной выдержки времени). Все эти состояния относятся к задачам «поставленным», т. е. к уже запущенным в произ-

водство деталям. Детали, покидающие по тем или иным причинам ГПС, соответствуют «решенным» задачам. Физический прообраз решаемой задачи — например деталь, перемещаемая на каретке свободной адресации от одной гибкой ячейки к другой для выполнения очередной технологической операции. Пример готовой задачи — описание детали, находящейся в приемопередающем устройстве гибкой ячейки, причем устройство ожидает команду на загрузку детали в рабочую зону станка.

Пример задержанной задачи — описание детали, непрерывность технологического процесса которой нарушена (из соображений оптимизации выполнения плановых заданий), деталь помещена в промежуточный накопитель или на склад. Пример ожидающей задачи — описание детали, находящейся в пристаночном накопителе гибкой ячейки. Примером приостановленной задачи-детали является та, которая не обеспечена необходимым инструментальным комплектом.

Каждой поставленной задаче (т. е. каждой запущенной в производство детали) соответствует структура данных фиксированного формата — дескриптор. Рассмотрим возможный набор дескрипторных данных.

Первая группа данных служит для идентификации задачи-детали по ее внешнему имени (коду). Внутренний идентификатор устанавливается системой оперативного управления при создании задачи.

Вторая группа данных — потребляемые ресурсы (инструменты, приспособления) и варианты технологических маршрутов.

Третья группа данных указывает состояние задачи (ставится, готова, задержана, приостановлена, ожидается). При этом для поставленной задачи зафиксирован код «процессора» (станка, транспортного средства). Для задач, пребывающих в других состояниях, есть коды причин, удерживающих задачи в этих состояниях.

В дескрипторные данные входят сведения о задачах (деталях), которые могут стать «преемниками» данной на «процессоре», а также информация для учета (циклограмма развития событий) и планирования (приоритет). Одновременно с решением задачи (т. е. одновременно с выпуском готового изделия) ее дескриптор уничтожается.

Ресурсы ГПС описываются по типу базы данных, доступ к которой обеспечивается системой управления базой данных.

Структура внешних связей системы оперативного управления (рис. 1) позволяет конкретизировать функциональное назначение системы (управление задачами и распределением ресурсов, организация ввода-вывода заготовок-изделий, обработка прерываний технологических «процессоров»).

Состояние входов можно описать циклограммой поступающего на обработку множества S партий деталей.

Состояние выходов можно описать циклограммой вывода из ГПС готовых деталей множества S .

Информация вектора освобождения деталей дает возможность построить циклограмму предполагаемого вывода готовых деталей из запланированного множества S . Сопоставление расчетной циклограммы с реальным появлением обработанных деталей на выходе позволяет судить о степени достоверности расписания и принять меры к его своевременной коррекции.

Рассмотрим внутреннюю структуру системы оперативного управления, отвечающую изложенным принципам. Отметим, что задача (т. е. деталь в процессе прохождения через ГПС) претерпевает многофазные переходы. Фазы соответствуют технологическим или транспортным операциям, переход от фазы к фазе можно прервать, задержать или приостановить.

Межфазные переходы отображаются графом (рис. 2), узлы которого соответствуют операциям, а дуги нагружены комплексными условиями межоперационных переходов. В каждом состоянии вызывается некоторая управляющая программа (на рис. 2 обозначена внизу

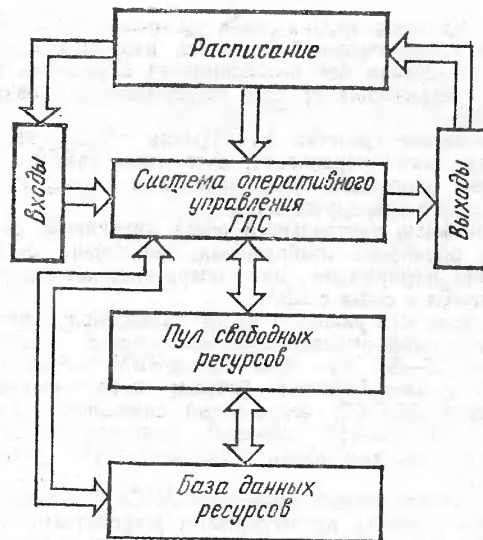


Рис. 1. Структура внешних связей системы оперативного управления ГПС

одела — как символ действия в транслирующей грамматике), обращенная к локальной микропроцессорной системе управления.

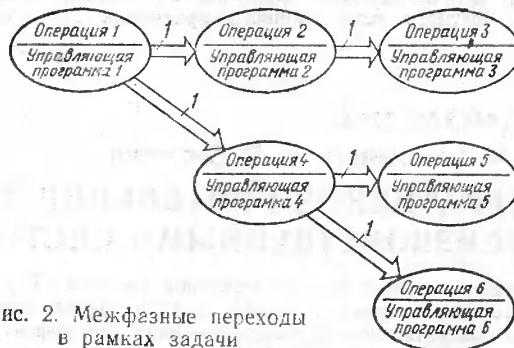


Рис. 2. Межфазные переходы в рамках задачи

Программный механизм, последовательно вызывающий управляющие программы, удобно строить по типу интерпретатора. Таким образом (наряду с параллельным решением задач, осуществляемым частью системы

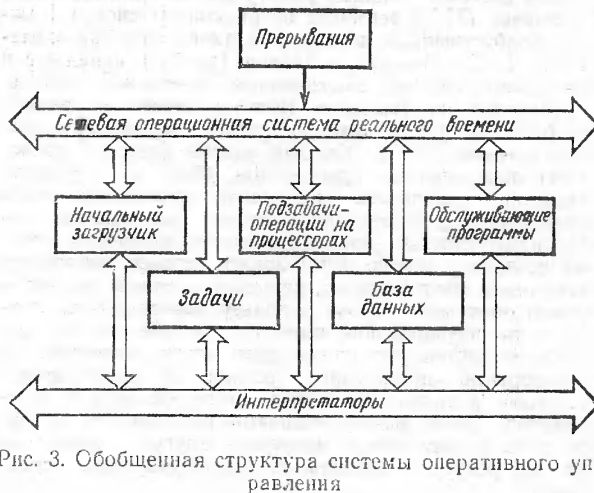


Рис. 3. Обобщенная структура системы оперативного управления

для каждой отдельной задачи дескрипторов использовать постоянно существующий дескриптор, описывающий всю динамику переходов для полного множества задач-деталей ГПС. Подобный подход учитывает то обстоятельство, что число «решаемых» в рамках ГПС задач-деталей примерно постоянно. В силу этого суммарный объем записей в дескрипторах достаточно стабилен, и этому объему может быть отведен фиксированный формат (точнее — набор фиксированных форматов соответственно переходам задач). Указанный дескриптор может быть изображен на абстрактном уровне в виде графа переходов (рис. 5).

Детализация отдельных переходов абстрактного графа и связей между ними приводит к структуре (рис. 6), переходами которой управляет система оперативного управления.

Макет двухуровневой системы создан в Московском станкоинструментальном институте на ЭВМ «Электроник»

ка 100—25» (являющейся машиной верхнего уровня в локальной сети микроЭВМ) в среде операционной системы РАФОС. Структура макета предполагает переход не только к безлюдному производству, но и безлюдному программированию технологических процессов в ГПС.

ЛИТЕРАТУРА

1. Системы управления станками.— Межвуз. сб. науч. тр. ВЗМИ, 1980, т. 14, с. 61—66.
2. Brandolese A., Garetli M. FMS control systems: Design criteria and performance analysis.— Proc. of the 2nd Int. Conf. on flexible manufacturing systems, 26—28 okt., 1983, London.— Bedford: IFS Publ., 1983, p. 365—381.

Статья поступила 25 июня 1985 г.

УДК 781.3:007.52

А. В. Каляев, Ю. В. Чернухин, Ю. А. Брюхомицкий, И. А. Каляев, В. П. Носков

ПОСТРОЕНИЕ ОДНОРОДНЫХ УПРАВЛЯЮЩИХ СТРУКТУР АДАПТИВНЫХ АВТОНОМНЫХ РОБОТОВ

Постановка задачи адаптивного управления. Развитие робототехники приводит к появлению задач, решение которых известными методами либо вообще невозможно, либо связано с существенными математическими и техническими трудностями. Одна из таких задач — оперативное управление адаптивным автономным транспортным роботом в динамически изменяющейся внешней среде. На рис. 1 показаны ситуации, возникающие при движении такого

робота к подвижной цели. В первый момент времени робот движется к цели по кратчайшей траектории 1, минуя все препятствия. Однако в процессе перемещения вдоль этой траектории перед ним возникает подвижное препятствие. Робот должен автоматически изменить траекторию движения 1 так, чтобы, минуя препятствия, достичь цель кратчайшим путем. Подобные изменения траектории движения он должен делать непрерывно, постоянно учитывая перемещения препятствий и перемещение цели. В приведенном примере роботу необходимо сначала изменить траекторию 1 на траекторию 2, а затем траекторию 2 на траекторию 3.

Более сложные ситуации возникают при функционировании робота в трехмерном пространстве (воздушная среда, космос, океан). Робот должен автоматически оценивать внешнюю обстановку, учитывать перемещение подвижных препятствий и одновременно решать задачу преследования, в общем случае, произвольно перемещающейся цели. Все эти действия необходимо выполнять с опережением реального масштаба времени.

Подобные ситуации возникают и при управлении адаптивными роботами-манипуляторами, функционирующими в естественной среде. Например, при управлении манипулятором подводного робота в процессе взятия каких-либо предметов из расщелин и трещин, робота-сборщика (рис. 2) и т. п. Если детали не точно попадают на одно и то же место рабочей зоны, меняются их размеры, то необходима адаптация манипулятора к этим изменениям, т. е. необходим поиск целевого положения среди препятствий.

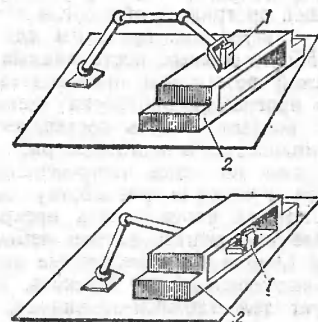


Рис. 2. Робот-манипулятор, выполняющий операцию установки детали 1 в полость детали 2

Наконец, сходные или еще более сложные задачи возникают при построении систем управления группами роботов, например в цехах при обслуживании транспортными роботами-разносчиками деталей для станков, в процессе сборки сложных деталей несколькими манипуляционными роботами-сборщиками и т. п.

Способ адаптивного управления. В рассмотренных случаях применим новый способ адаптивного управления, основанный на использовании однородных управляющих структур (ОУС). Его суть заключается в том, что система управления роботом в процессе динамического взаимодействия с априори неформализованной внешней средой периодически строит модель последней, которая затем воспроизводится в состояниях ОУС в виде свободных и запрещенных для движения участков, а также участков, соответствующих исходному состоянию робота и цели.

В каждом цикле работы системы от участка, характеризующего цель, в ОУС распространяется волна воз-

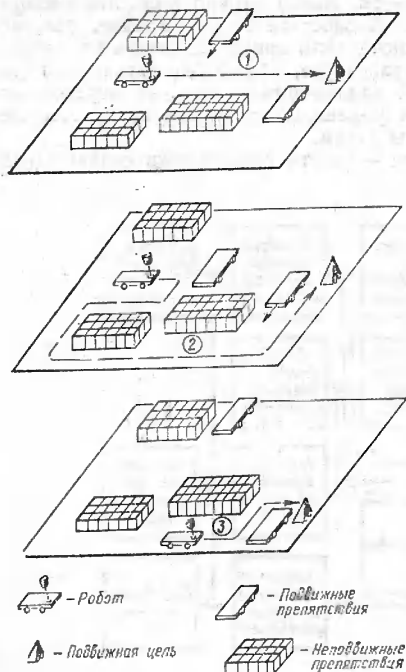


Рис. 1. Движение транспортного робота к подвижной цели

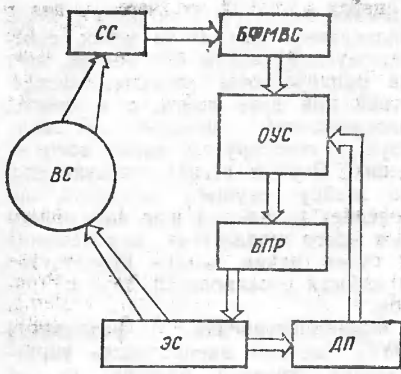


Рис. 3. Структурная схема управляющей системы адаптивного робота

буждения, которая «строит» различные возможные траектории движения робота к цели. Начальный участок траектории, оказавшейся оптимальной в данный момент времени, фиксируется в ОУС и затем отрабатывается в виде элементарного шага эффекторной системы робота. В результате робот переходит в новое состояние. Циклы работы системы повторяются с учетом изменений, происходящих во внешней среде и в местоположении робота до тех пор, пока он не достигнет цели.

В отличие от известных искома траектория в предлагаемом способе формируется не путем последовательного отбрасывания неудачных эвристик, а в результате одновременной генерации возможных гипотез достижения цели и выделения наиболее удачной из них. Причем фиксация программы будущего движения непосредственно сопровождается ее реализацией в последовательных состояниях самого управляемого объекта.

ОУС адаптивного робота. Управляющая система робота (рис. 3), построенная по изложенному способу, функционирует следующим образом. Информация о внешней среде (ВС) через сенсорную систему (СС) поступает в блок формирования модели внешней среды (БФМВС), отображающий ее в со-

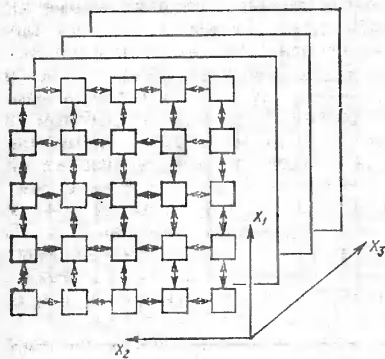


Рис. 6. Архитектура 3-мерной ОУС

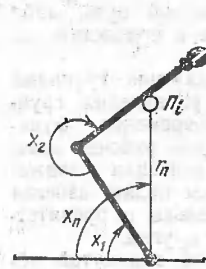


Рис. 4. Кинематическая схема манипулятора, представленного плоским двухзвенником:

X_1, X_2 — угловые координаты звеньев манипулятора; $X_{Пi}$ — полярные координаты точечного препятствия $П_i$

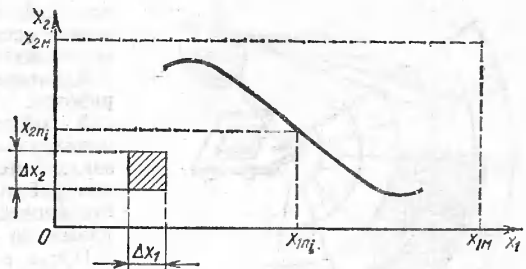


Рис. 5. Образ точечного препятствия в фазовом пространстве состояний двухзвенного манипулятора: X_{1m}, X_{2m} — максимальные значения угловых координат звеньев манипулятора

стояниях ячеек ОУС. Из ячейки, соответствующей конечному состоянию (состоянию достигнутой цели), строится дерево возможных способов достижения этого состояния. Одна из ветвей дерева попадает в ту ячейку ОУС, которая характеризует состояние объекта в данный момент. Блок принятия решений (БПР) выделяет направление этой ветви и формирует сигнал управления, отрабатываемый эффекторной системой (ЭС). С помощью датчиков положения (ДП) определяется состояние объекта в данный момент времени. После перехода управляемого объекта в новое положение, указанные действия повторяются.

Связи элементов ОУС и функции БФМВС зависят от назначения робота. Если объектом управления является робот-манипулятор (рис. 4), то БФМВС отображает объекты внешней среды в пространство состояний манипулятора, т. е. фазовое пространство, каждая точка которого соответствует некоторой конфигурации манипулятора. Поэтому каждой точке внешней среды соответствует множество точек фазового пространства (рис. 5). Например, если в двумерном рабочем пространстве плоского двухзвенника есть точечное препятствие $П_i$, то соответствующий ему образ в фазовом пространстве состояний манипулятора имеет вид некой кривой.

Однородная управляющая структура представляет собой сеть логических элементов (рис. 6). В случае плоского двухзвенника достаточно иметь только одну плоскость $X_1 X_2$. Для трехзвенного манипулятора необходима 3-мерная структура $X_1 X_2 X_3$ и т. д. При этом каждая ячейка n -мерной управляющей структуры остается очень простой (рис. 7). Функционирование такой ячейки зависит от информации, поступающей из внешней среды. В ячейке, соответствующей разрешенному положению манипулятора, из БФМВС на вход Запрет посту-

пает сигнал «Лог. 1», в противном случае на этот вход поступает сигнал «Лог. 0». Если это положение манипулятора является целевым, то сигнал «Лог. 1» поступает на входы Цель и Запрет данной ячейки одновременно. Далее ячейка начинает работать в режиме генератора единичных сигналов, распространяющихся по ОУС только через открытые ячейки. Сигнал «Собственное положение» (СП) поступает от ДП и предназначен для выделения той ячейки ОУС, которая соответствует в данный момент времени положению схвата манипулятора.

Блок принятия решения состоит из таких же ячеек, включенных по схеме n -стабильного триггера. БПР выделяет и фиксирует ту ветвь сети, по которой сигнал возбуждения приходит первым.

Результаты эксперимента. Для проверки рассмотренного принципа адаптивного управления разработана управляющая система трехзвенным роботом-манипулятором, имеющим пять степеней подвижности. ОУС состоит из 960 ячеек ($20 \times 16 \times 3$), размещенных на 60 платах, каждая из которых, в свою очередь, содержит 90 микросхем серии К-134 (всего около 6000 микросхем). Об-

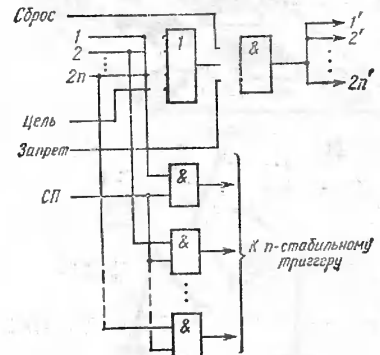


Рис. 7. Функциональная схема ячейки ОУС

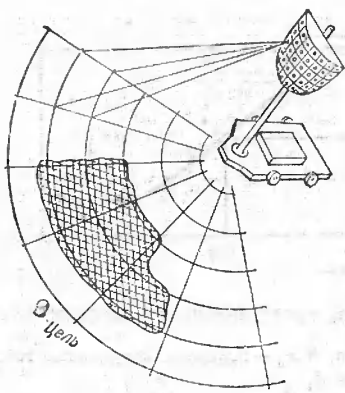


Рис. 8. Отображение внешней среды на элементы ОУС транспортного робота

ный объем устройства — 0,08 м³ (см. с. 3 обложки).

Аналогичный принцип управления использован в лабораторном макете адаптивного транспортного робота (см. с. 3 обложки). Внешняя среда робота в виде светлых, темных и ярких участков отображается через дистантные датчики (в макете это фотодатчики) на элементы ОУС. Темные участки воспринимаются роботом как препятствия, светлые — как свободные для движения пути, яркий участок — как цель движения. Задача робота заключается в движении к цели по оптимальному (кратчайшему) пути, минуя препятствия (рис. 8).

По окончании приема информации о среде, в ОУС генерируются волны возбуждения, моделирующие возможные варианты поведения, т. е. еще до начала движения робот «мысленно продумывает» будущую стратегию своего поведения и выбирает «с его точки зрения» более оптимальную. «Думает» робот и в процессе движения. Если на выбранном им пути неожиданно возникают препятствия, он меняет стратегию поведения. Например, он может выбрать другой, менее опти-

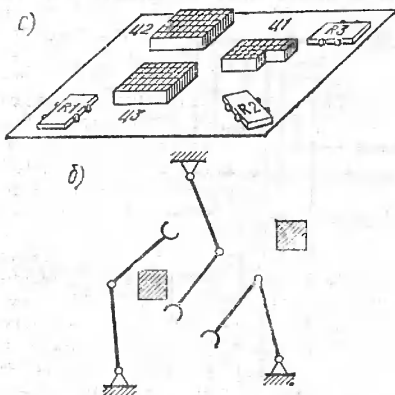


Рис. 9. Группы транспортных (а) и манипуляционных (б) роботов

мальный, но свободный путь, либо может остановиться и подождать до освобождения трассы.

Адаптивное управление группами роботов. Задача управления группой роботов, одновременно функционирующих в одной рабочей зоне, заключается в организации движения роботов к своим целям, избегая столкновения не только с препятствиями, но и друг с другом.

Пусть в некоторой замкнутой области с препятствиями функционирует группа транспортных роботов, причем перед каждым роботом поставлена своя задача перемещения в целевую точку (рис. 9). Разобьем рабочую зону на множество непрерывно соединяющихся участков, соизмеримых с размерами роботов, а под состоянием группы будем понимать некоторое расположение роботов в такой дискретной области. Каждому возможному состоянию группы роботов поставим в соответствие ячейку ОУС. Связи между ячейками должны определять возможные переходы группы из одного состояния в другое. Кроме того, заблокируем в ОУС множество ячеек, соответствующих запрещенным состояниям группы, т. е. таким состояниям, в которых хотя бы один робот попадает в зону расположения препятствий, либо более одного робота находится в одном дискретном участке. В результате в ОУС будет сформирована модель пространства состояний группы транспортных роботов.

Генерация волны возбуждения из ячейки, соответствующей целевому состоянию группы и фиксации первого из сигналов, поступивших на ячейку текущего состояния, позволяет определить начальный шаг наилучшего «с точки зрения» всей группы перемещения роботов к заданным целям. Циклическое повторение такой процедуры организует движение роботов к своим целям, минуя препятствия и избегая столкновений друг с другом. Высокое быстродействие ОУС обеспечивает функционирование системы в реальном масштабе времени. Это открывает широкие перспективы для применения ОУС при создании робототехнических транспортных систем, обслуживающих производственные цеха, склады и т. п.

Аналогичный подход может быть использован и при решении задачи управления группами манипуляционных роботов, участвующих, например, в едином сборочном процессе. Здесь под состоянием группы роботов следует понимать некоторый набор дискретных конфигураций манипуляторов, входящих в группу. Как и ранее, каждому состоянию роботов необходимо поставить в соответствие ячейку ОУС, а связи между ячейками будут определять возможные переходы роботов из одного со-

стояния в другое. При этом в ОУС блокируется множество ячеек, соответствующих таким состояниям, когда манипуляторы касаются препятствий или друг друга, а в ячейке, определяющей целевое состояние группы, генерируется волна возбуждения. Первый сигнал, поступивший на ячейку текущего состояния, определяет начальный шаг наилучшего для всего коллектива перемещения к своим целям, минуя препятствия и избегая столкновений друг с другом.

Микроэлектронная реализация ОУС. Высокая однородность управляющих структур, простота их отдельных ячеек создают благоприятные предпосылки для изготовления фрагментов и даже структур в целом в виде больших и сверхбольших интегральных схем. Это позволило НИИ Таганрогского радиотехнического института приступить к разработке БИС ОУС для систем управления адаптивными роботами. ОУС имеет плоскую гексагональную архитектуру и состоит из 384 (32 × 12) ячеек, каждая из которых двунаправленными связями соединена с шестью соседними ячейками. Такая архитектура позволяет передавать волны возбуждения по шести направлениям, что хорошо согласуется с радиальной системой обзор транспортного робота.

В структурной схеме БИС (рис. 10) предусмотрена возможность компоновки необходимой размерности ОУС путем сращения нескольких аналоговых БИС. Адресация ячеек БИС ОУС производится по двум координатам, которые в прямоугольной структуре соответствуют непосредственно ее прямоугольным координатам, а в радиальной структуре, характерной для управляющих систем транспортных роботов, полярным координатам φ, R.

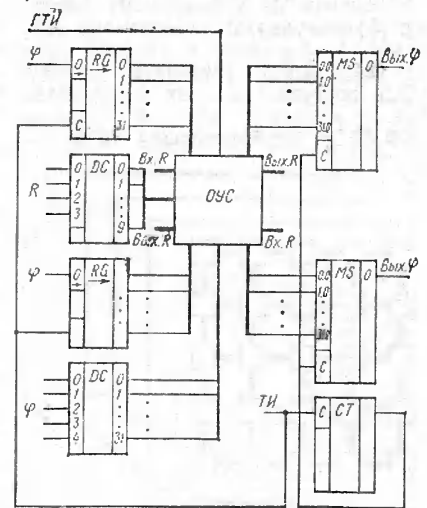


Рис. 10. Структурная схема БИС ОУС

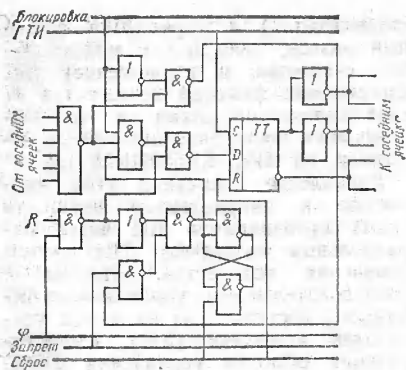


Рис. 11. Функциональная схема ячейки БИС ОУС

В режиме настройки с помощью дешифраторов выбираются ячейки ОУС, соответствующие запрещенным участкам траектории движения робота. Мультиплексоры и регистры организуют последовательную связь с соседними аналогичными БИС (при их наличии) в условиях существующих ограничений на число выводов БИС. При этом мультиплексоры преобразуют параллельный код состояний крайних ячеек ОУС по координатам Φ , R в последовательный, которым возбуждение передается через выходы данной БИС на входы соседних. Регистры производят обратное преобразование последовательного кода, поступающего из соседних БИС, в параллельный, передаваемый на крайние ячейки данной БИС. Двоичный счетчик управляет мультиплексорами. Работа БИС синхронизирована сериями так-

УДК 681.325.5 : 007.52

Ю. С. Смирнов

СИСТЕМЫ УПРАВЛЕНИЯ СЕРВОМЕХАНИЗМАМИ С ШАГОВЫМИ ЭЛЕКТРОДВИГАТЕЛЯМИ

Широкое внедрение микропроцессоров (МП) сдерживается недостаточным развитием периферийной техники [1]. Примером в этом отношении являются сервомеханизмы (СМ) роботов, для сопряжения которых с МП необходимы преобразователи «код — угол» и «угол — код» [2]. При создании этой ветви периферийной техники для МП необходимо предусматривать разработку заказных и полузаказных интегральных устройств для аналого-цифровой обработки и преобразования измерительных и управляющих сигналов на основе инженерных решений, удовлетворяющих ограничениям по стоимости, надежности и удобству эксплуатации [3].

Особенности построения и взаимодействия комплекса «МП—СМ» опре-

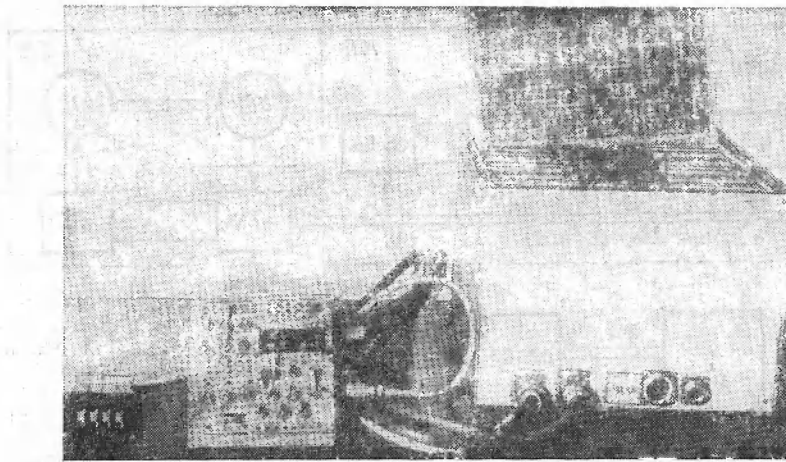


Рис. 12. Макетный образец системы управления транспортным роботом

товых и главных тактовых импульсов (ТИ и ГТИ). В схеме ячейки (рис. 11) предусмотрен анализ наличия ложного возбуждения.

БИС ОУС предполагается выполнить на основе одного базового кристалла по КМОП-технологии в стандартном корпусе с 64 выводами. Уровень интеграции при этом составит более 35 тыс. транзисторов. Высокий уровень интеграции, большой процент выхода годных кристаллов, повышенную надежность и живучесть БИС планируется достигнуть за счет регулярности ее структуры, позволяющей производить отбраковку и последующую блокировку неисправных ячеек БИС непосредственно в процессе ее изготов-

ления путем проверки всех ячеек ОУС на специальной контрольно-испытательной аппаратуре.

Использование БИС ОУС в системах управления адаптивными транспортными роботами позволит существенно улучшить их массо-габаритные характеристики, повысить надежность и живучесть, снизить стоимость. В частности, в системе управления адаптивным транспортным роботом, разработанной в НИИ Таганрогского радиотехнического института (рис. 12), применение только одной БИС ОУС позволит заменить комплект аппаратуры объемом 0,027 м³, содержащий 1500 микросхем серии К164.

Статья поступила 22 января 1985 г.

нагрузке шага, а быстрдействие — частотой обработки ШЭД в режиме пошагового управления. Частота эта зависит от нагрузки и обычно не превышает 1000 Гц, что ограничивает применение такого ШЭП в СМ высокой точности и производительности.

Микропроцессорное управление СМ с ШЭД повышает эффективность как разомкнутого, так и замкнутого ШЭП. Использование разомкнутого ШЭП позволяет обойтись без преобразователя «угол — код», стоимость отдельных вариантов которого сопоставима со стоимостью простого манипулятора [2]. Применение исполнительных элементов с разветвленной кинематической схемой, раздельными прецизионными датчиками положения и скорости помимо увеличения стоимости приводит к усложнению эксплуатации и снижению надежности системы. Замена сложного непрерывного привода более простым разомкнутым ШЭП в сочетании с пакетом программ позволяет получить существенный выигрыш в его стоимости [7]. Инвариантность разомкнутого ШЭП от изменений на-

деляются типом исполнительного элемента. Наибольшей совместимостью с МП обладает шаговый электропривод (ШЭП) в его замкнутом и разомкнутом вариантах, осуществляющий преобразование «код — угол» [4].

Схема управления замкнутым ШЭП от МП представлена на рис. 1, а. Шаговый электродвигатель (ШЭД) работает в режиме пошагового управления. Контроль обработки шагов производится через цифровой преобразователь угла (ЦПУ) [5], осуществляющий преобразование «угол — код». Сопряжение СМ с МП производится через таймер (Т), который может обеспечить параллельную работу нескольких каналов [6]. Разрешающая способность СМ определяется дискретностью приведенного к

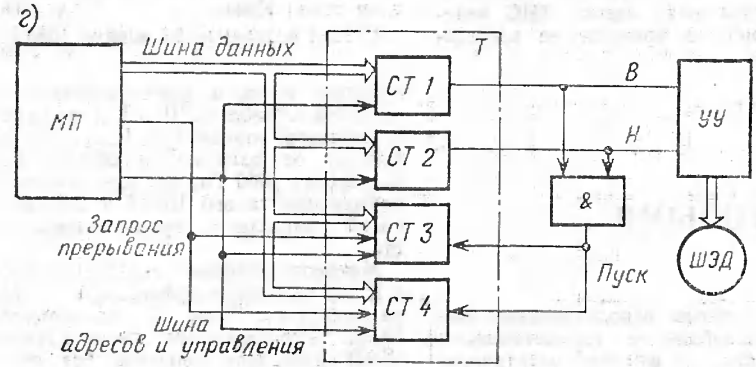
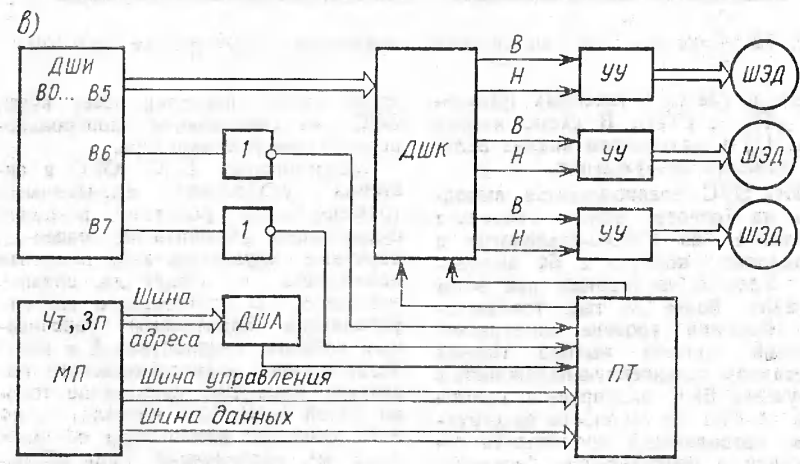
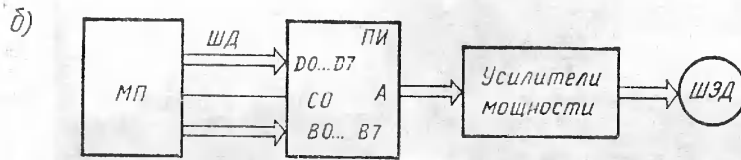
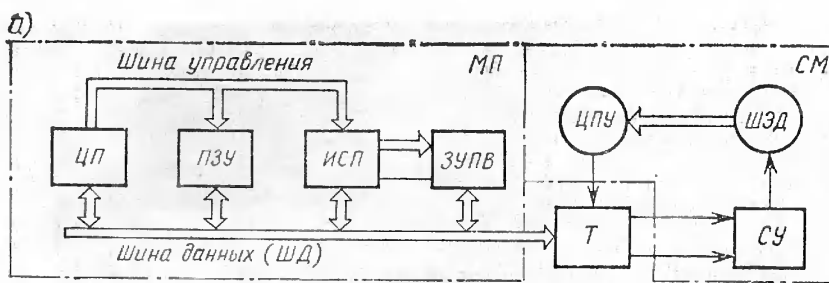


Рис. 1. Сопряжения микропроцессора с замкнутым ШЭП через интервальный таймер (а), разомкнутым ШЭП через параллельный программируемый интерфейс (б), многодвигательным ШЭП через программируемый таймер (в) разомкнутым ШЭП через программируемый контроллер (г):

ЦП — центральный процессор; ПЗУ — постоянное запоминающее устройство; ИСП — интерфейс связи с памятью; ЗУПВ — запоминающее устройство с произвольной выборкой; СУ — схема управления

грузки должна обеспечиваться программно, а эффективным средством повышения его производительности служит программируемый разгон с оптимальным нарастанием и уменьшением частоты управления [6]. Это позволяет вести устойчивую обработку больших перемещений на частоте,

в 2—3 раза превышающей частоту пошагового управления. Для упрощения электронной части системы наибольший интерес представляет микропроцессорная система управления (рис. 1, б), совмещающая задания алгоритмов функционирования (позиционирование, слежение или

стабилизация) и управления (плавный разгон, движение с максимальной скоростью и торможение) [8]. Совмещение функций достигается за счет программирования и соответствующего построения интерфейса, например, на БИС КР580ИК55 [9].

Управление величиной угла, скоростью и направлением вращения ШЭП производится МП через параллельный интерфейс (ПИ) путем изменения количества, частоты и последовательности управляющих импульсов, поступающих на входы усилителей мощности (УМ), коммутирующих обмотки управления ШЭД. Порт А параллельного интерфейса выполняет функции распределителя импульсов устройства управления (УУ) ШЭД. Значение временного интервала между соседними импульсами в каждом канале, т. е. частота импульсов управления и соответствующая ей скорость вращения ШЭД, программируется посредством порта В. Направление вращения ШЭД задается на выводе С₀ порта С. В зависимости от логического значения этого сигнала происходит циклический сдвиг влево или вправо на разрядах порта А. Недостатком такого построения является снижение помехоустойчивости в канале коммутации ШЭД при управлении другими процессами посредством прерывания [10].

Существенно снижение аппаратных затрат в системе с многодвигательным ШЭП можно достигнуть некоторым усложнением контроллера (рис. 1, в), обеспечивающего управление одним МП нескольких ШЭД. Режим работы трех счетчиков программируемого таймера (ТТ) БИС КР580ВИЗ [9] задается процессором КР580ИК80. При управлении одним ШЭД первый счетчик выполняет функции генератора импульсов, а второй — одновибратора, подсчитывающего их число. Вентили дешифрагора каналов (ДШК) стробируются МП через порт В дешифрагора (адаптера) интерфейса (ДШИ). В случае двух двигателей третий счетчик служит одновибратором.

Контроллер позволяет независимо изменять скорость вращения и угол поворота. Это достигается тем, что во втором счетчике загружено число импульсов для ШЭД, совершающих большее число шагов, чем в третьем. Запуск их производится одновременно, и при обнулении третьего счетчика через ДШК отключается УУ соответствующим ШЭД. Программируемый разгон ШЭД осуществляется многократным перезапуском третьего счетчика при непрерывной работе второго. Форма кривой разгона определяется регулирующими приращениями частоты, загруженными в третий счетчик. Вращение контролируется ШЭД в режимах слежения и прерывания [6].

Когда управление производится с

разделением во времени и недопустимы дополнительные накладные затраты на «дежурные» временные циклы и программы, возможно иное построение интерфейса (рис. 1, г), предусматривающее управление ШЭД от МП через программируемый контроллер. Помимо повышения помехоустойчивости такое построение упрощает задачу составления программ для интеллектуального интерфейса. Счетчики СТ1 и СТ2 таймера служат генераторами тактовых импульсов соответственно «вперед» (В) и «назад» (Н), а СТ3 подсчитывает их количество. Чтобы уменьшить накладные затраты программных средств при разгоне и торможении, задействован счетчик СТ4, подтверждающий команду разгона и выдающий сигнал прерывания, показывая, что контроллер готов для приема следующей инструкции. Реакцией МП на эту ситуацию является загрузка новой константы в СТ1 и СТ2, т. е. изменение скорости ШЭД. В программе предусматривается использование высокоуровневых команд для управления ШЭД [10].

Благодаря этому интерфейсу и программным средствам другими процессами реального времени можно управлять в фоновом режиме с помощью прерываний, что практически не влияет на расстановку управляющих импульсов. Однако для таких процессов необходимо выбрать соответствующие интервалы разгона ШЭД, что может отразиться на производительности ШЭП. Это, наряду с ограниченной помехозащищенностью, сужает область применения СМ, в которых ШЭД работает в разомкнутом ШЭП с программируемым разгоном.

Существенный недостаток разомкнутого ШЭП — возможность невосполнимой потери информации при преобразовании выходных сигналов МП в перемещения. Применение в СМ для контроля этого преобразования ЦПУ накопительного типа (см. рис. 1, а) целесообразно в устройствах средней точности и производительности. Основным недостатком как разомкнутого ШЭП, так и замкнутого с обратной связью на основе накопительного ЦПУ является отсутствие информации о текущей координате относительно исходной точки начала движения, что требует калибровки СМ перед каждым включением.

При работе СМ в условиях случайных воздействий ШЭП должен быть замкнут по положению. Это обеспечивает парирование внешних и внутренних возмущений. Для формирования МП оптимальных и адаптивных алгоритмов управления в системе автоматического управления необходимо иметь цифровой эквивалент скорости [2]. Современная робототехника требует применения ЦПУ с точностью 12...16 бит и бы-

стродействием $10^9 \dots 10^{10}$ преобразований в секунду [3]. В канале скорости должно произойти преобразование частот вращения $0,01 \dots 100$ % с информационной емкостью 10...14 бит. Эти значения должны обеспечиваться во всем диапазоне воздействия внешних и внутренних дестабилизирующих факторов.

Одним из путей решения этой задачи является разработка *специальных кодовых датчиков* с использованием оптических и электромагнитных элементов в сочетании с электронными устройствами считывания и преобразования выходной информации [5]. К их достоинствам следует отнести высокое быстродействие, простоту сопряжения с шинами МП и возможность получения цифрового эквивалента скорости методом цифрового дифференцирования кода угла. Однако необходимые вычисления требуют значительных затрат машинного времени [3] и не обеспечивают оперативного получения кода скорости в широком диапазоне частот вращения СМ. Эти ограничения, наряду с высокой стоимостью [2], пониженной устойчивостью к воздействию внешних факторов и ряда конструктивных ограничений, заставляют искать иные пути построения ЦПУ.

Для робототехнических систем актуально создание *измерителя параметров движения с одним первичным датчиком* [11], обеспечивающим в сочетании с микроэлектронными преобразователями получение цифровых эквивалентов угла и скорости. Эта задача решается построением ЦПУ, в которых в качестве первичного датчика используется синусно-косинусный вращающийся трансформатор (СКВТ), преобразующий угол в

электрический сигнал. Информация об угловом положении вала в СКВТ представлена в виде модулированных сигналов переменного тока. Освоена и выпускается широкая гамма контактных и бесконтактных СКВТ. Они удовлетворяют жестким эксплуатационным требованиям, имеют высокий срок службы и надежность. По метрологическим показателям они не уступают кодовым датчикам, а по информативности при изменении скорости превосходят их. Стоимость преобразователей «угол — параметр — код» на основе СКВТ, с учетом электронной отсчетной части, меньше, чем стоимость кодовых датчиков с сопоставимыми размерами и выходными параметрами.

По методу получения цифрового эквивалента угла преобразователи «угол — параметр — код» подразделяются на *циклические* и *следающие*. В качестве промежуточного параметра в них используют фазу или амплитуду выходных сигналов СКВТ. При соответствующем построении амплитудный преобразователь сопоставимых габаритов и точности имеет преимущества перед фазовыми в части повышения стабильности в диапазоне температуры, не критичности к частоте питающего напряжения датчика и его нелинейным искажениям [11]. Это в равной степени относится как к преобразователям циклическим, так и следающим.

Существенное достоинство циклических преобразователей — простота их сопряжения с МП через ПИ (рис. 2), например КР580ИК55. В процессе получения координат при воспроизведении сложных траекторий такие преобразователи обеспечивают дополнительные преимущества. При

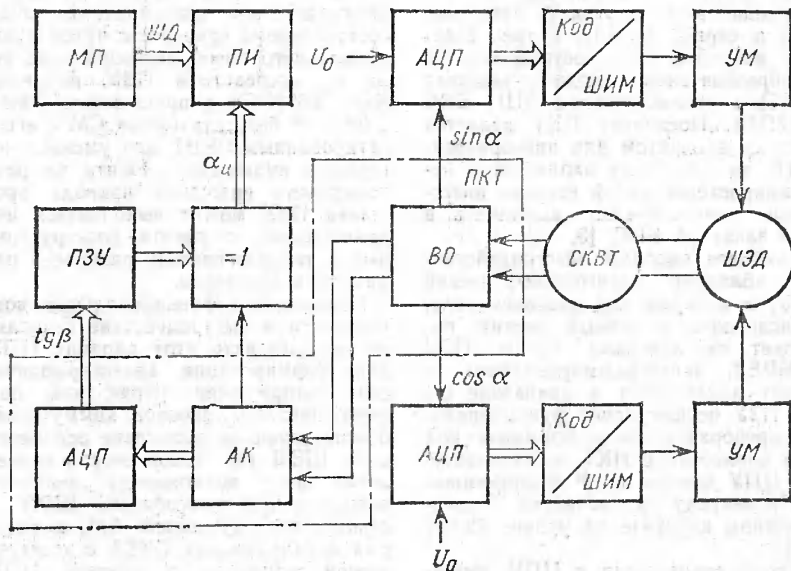


Рис. 2. Сервомеханизм с миннишаговым устройством управления:

АК — аналоговый коммутатор, ВО — выявитель октантов, ШД — шина данных

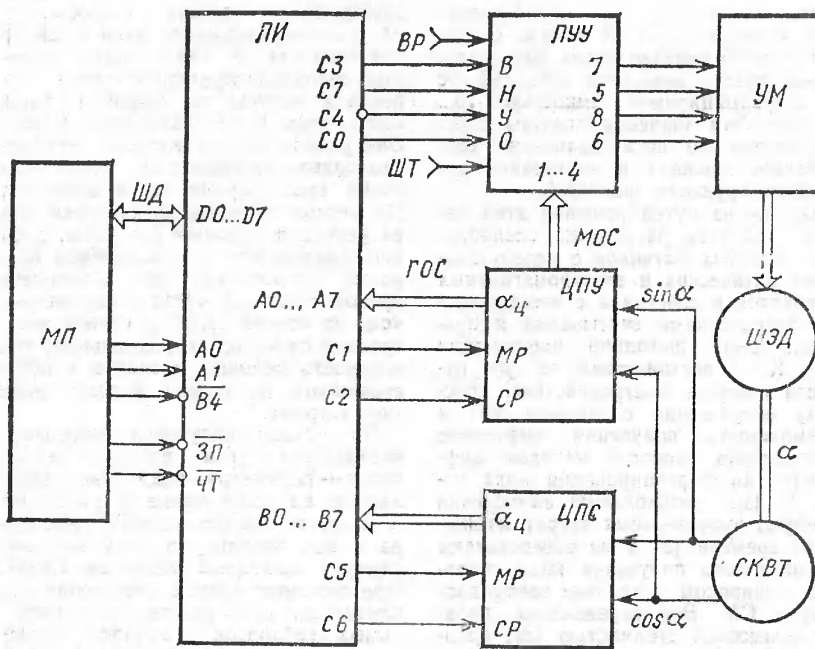


Рис. 3. Сервомеханизм с ШЭД в режиме самокоммутации:

ВР — выбор режима; ШТ — шина тактов; ПУУ — полирежимное устройство управления (У — ускорение вращения, О — останов, 1...4 — входы МОС, 5...8 — выходы распределителя ПУУ); МР — младшие разряды; СР — старшие разряды

соответствующем построении отсчетной части [12] экономится машинное время микроЭВМ за счет вычисления функциональным ЦПУ не только цифрового эквивалента угла α_d , но и кодов его ортогональных проекций $\sin \alpha$ и $\cos \alpha$. Для работы с МП предложены три разновидности амплитудных циклических ЦПУ: функциональный, масштабирующий и тригонометрический, в которых полученные коды тангенса угла β , приведенного в первый октант, и трех старших разрядов кода осуществляется преобразователем кода тангенса (ПКТ) с использованием АЦП БИС К572ПВ1. Поскольку ПКТ является базовым элементом для преобразователей, то остальные входящие в него микросхемы малой степени интеграции целесообразно выполнять в виде заказной БИС [3, 13].

Наиболее высоким быстродействием обладает тригонометрический ЦПУ, в котором код тангенса угла, приведенного в первый октант, поступает на адресные входы ПЗУ К505РЕЗ, запрограммированного по закону арктангенса в диапазоне 0...45°. ПЗУ осуществляет функциональное преобразование и формирует код угла совместно с ПКТ. Быстродействие ЦПУ близко к 10^4 преобразований в секунду при точности в одноотсчетном варианте на уровне 12 бит [14].

Путем сопряжения с ЦПУ дополнительных БИС просто реализуется функциональный преобразователь с повышенным быстродействием. При умеренных требованиях к точности и

скорости формирования кодов проекций они могут быть получены в режиме временного разделения на одной отсчетной части с кодом угла α_d . Если определяющим является быстродействие, то введением двух дополнительных АЦП БИС К572ПВ1 (см. рис. 2) реализуется параллельное преобразование каждой составляющей. В случае, когда необходимо совместить высокую точность и быстродействие, формирование кода составляющих производят путем функционального преобразования кода угла α_d посредством ПЗУ, например БИС К505РЕЗ с прошивками 0068...0071. В безредукторных СМ с агрегатированным СКВТ для уменьшения влияния пульсаций момента на равномерность вращения привода прошивка ПЗУ может выполняться индивидуально, с учетом конструктивных и технологических разбросов параметров двигателя.

Повышенные функциональные возможности и быстродействие позволяют использовать этот вариант ЦПУ для формирования квазигармонических напряжений управления при «миншаговом» способе коммутации, обеспечивающем дробление основного шага ШЭД [4]. Такой способ управления дает возможность повысить разрешающую способность ШЭП и строить безредукторные СМ, в которых двухканальный СКВТ с электрической редукцией в составе ЦПУ обеспечивает формирование сигналов управления и получение цифрового эквивалента перемещения относительно начальной точки отсчета с инфор-

мационной емкостью 14...16 бит в диапазоне одного оборота ротора многополюсного двигателя.

Сигнал управления на усилитель мощности ШЭД подается через преобразователи «Код — ШИМ» (см. рис. 2). Это повышает быстродействие и точность перед построением, использующим преобразование «угол — фаза — код» [4]. Недостаток миншагового управления: производительность ШЭД остается неизменной по сравнению с пошаговым управлением.

Эффективность замкнутого ШЭП существенно возрастает при переводе ШЭД в режим самокоммутации по сигналам местной обратной связи (МОС), для формирования которой в безредукторном СМ используются соответствующие разряды ЦПУ [15] с СКВТ в качестве первичного датчика (рис. 3). Сопряжение СМ с МП производится через ПИ, например БИС КР580ИК55 [9]. Управление преобразователями осуществляется через порт С, так как каждый разряд его регистра может быть установлен в состояние «1» или «0» одной командой. Прием данных производится через порты А и В. Такая структура ШЭП позволяет упростить СМ, повысить его надежность и быстродействие за счет исключения дополнительных элементов, связанных с валом ШЭД. Производительность СМ возрастает на порядок по сравнению с пошаговым управлением. В связи с этим повышаются требования к производительности ЦПУ.

Дальнейшее повышение производительности ЦПУ ограничено не только конечным быстродействием электронных компонентов, но и принципиальными ограничениями, связанными с методом поразрядного кодирования в циклических ЦПУ. Поэтому для преобразования быстрых угловых перемещений используются следующие ЦПУ, обеспечивающие меньшие динамические ошибки. Этот тип преобразователя менее чувствителен к гармоникам, шумам и квадратурным составляющим выходных сигналов СКВТ. Такие преобразователи получили значительное распространение и за рубежом [5].

При сопряжении следящих ЦПУ с МП возникает ряд трудностей: их невозможно подключить непосредственно к его магистралям; сьем данных в динамическом режиме затруднителен и может привести к грубым ошибкам, а останов для сьемки данных нежелателен в быстродействующих системах, так как следящие ЦПУ в отличие от циклических при включениях имеют значительное время установки, снижающее частоту их обмена с МП. Эти трудности устраняются при введении в состав ЦПУ дополнительного «прозрачного» регистра и выходных каскадов с тремя состояниями, организованных как два независимо включаемых байта

[13]. Команду фиксации можно подать в любое время благодаря логическому исключению возможности искажения информации из-за совпадения моментов смены и фиксации данных в регистре. Поэтому возможен съём данных без прерывания слежения и сопряжения по принципу распределенной памяти и управления с большинством 8- и 16-разрядных МП.

Выбор оптимального типа ЦПУ для конкретного применения — трудная задача, усложняемая рядом противоречивых ограничений, связанная с характеристиками циклических и следящих ЦПУ. Компромиссным является ЦПУ с переменной структурой [16]. В установившемся состоянии он работает в режиме слежения, а при переходных процессах — как циклический. Алгоритм его работы меняется блоком переключения режимов (БПР) в зависимости от ошибки преобразования (рис. 4). ЦПУ является замкнутой электронной следящей системой, в которой контур регулирования содержит: последовательно включенный фазочувствительный выпрямитель (ФЧВ); генератор, управляемый напряжением (ГУН); двоичный счетчик; схему Исключающее ИЛИ; ПЗУ с тангенсной прошивкой; формирователь кода тангенса — котангенса (ФКТК) и дифференциальный усилитель (ДУ). Если напряжение рассогласования превышает уровень компараторов, то система из режима пропорционального управления переводится в релейный режим с поразрядным уравниванием рассогласования. Введе-

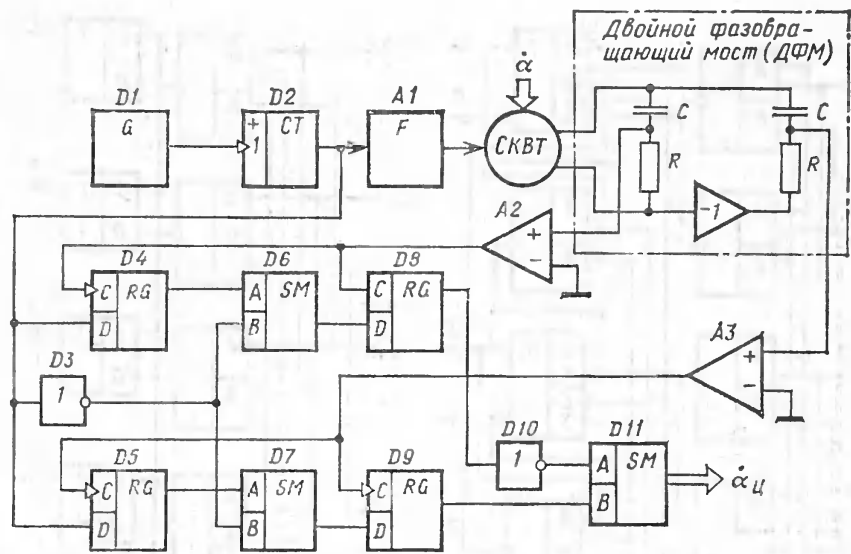


Рис. 5. Цифровой преобразователь скорости

ние БПР с компараторами и дополнительным регистром позволяет не только решить задачу сопряжения ЦПУ с МП, но и более чем на два порядка сократить время установки кода на выходе ЦПУ по сравнению с его следящим вариантом. Это особенно важно для оптимальных систем с ШЭД в режиме самокоммутации.

Код скорости можно получить преобразованием сигнала рассогласования АЦП К572ПВ1 (см. рис. 4) либо от цифрового преобразователя

скорости (ЦПС) [17], схема которого представлена на рис. 5. В отличие от известных построений [18] в нем используется режим работы СКВТ с пульсирующим полем, а выходные сигналы СКВТ подаются на вход отсчетной части ЦПС через ДФМ. Это позволяет простыми средствами [19] преодолеть основной недостаток ЦПС на СКВТ с вращающимся полем [18], заключающийся в невозможности одновременного обеспечения высокой точности измерения углового положения и скорости вращения СКВТ в широком диапазоне ее изменения. Разность периодов выходных напряжений моста с высокой степенью точности пропорциональна скорости изменения угла [19]. Цифровая часть ЦПС (D1—D11) реализуется на микросхемах серии К155, ее функции могут выполняться МП по соответствующей программе. Применение СКВТ с электрической редукцией позволяет повысить точность и расширить диапазон измерения угла и скорости.

Следует отметить, что при сопряжении таких ЦПУ и ЦПС в микропроцессорных системах реального времени необходимы повышенные требования к быстрдействию интерфейса. В комплексе МП—СМ возможна организация считывания данных с обращением к преобразователям как к ячейке памяти или как к внешнему устройству. Для МП К580ИК80 время выполнения программы записи при разрядности преобразователя 2 байта для обоих вариантов обращения равно 38 мкс. Время на организацию запуска при программном и с прерыванием инициировании опроса составляет соответственно 63 и 112 мкс. При более высокой частоте ввода информации от СМ в память МП следует исполь-

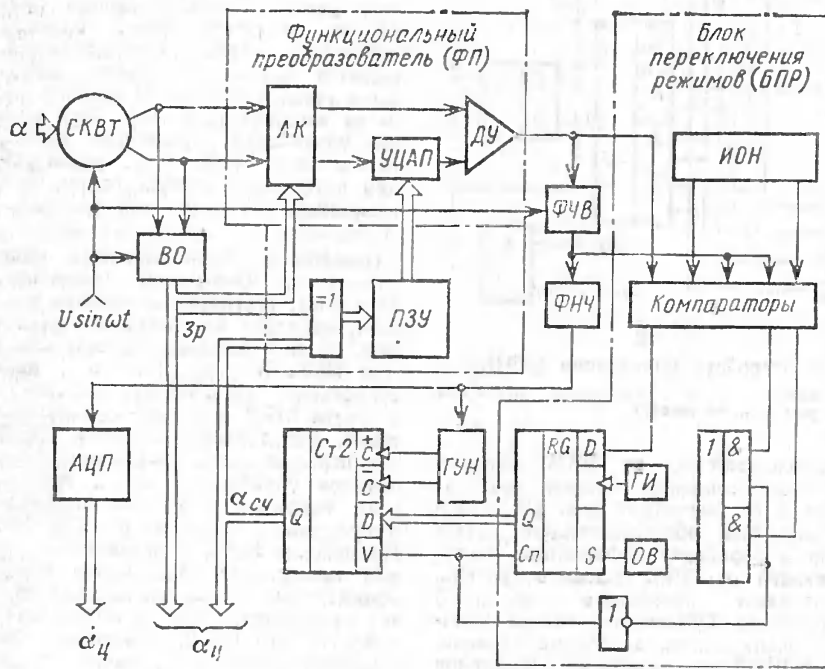


Рис. 4. Цифровой преобразователь угла с переменной структурой

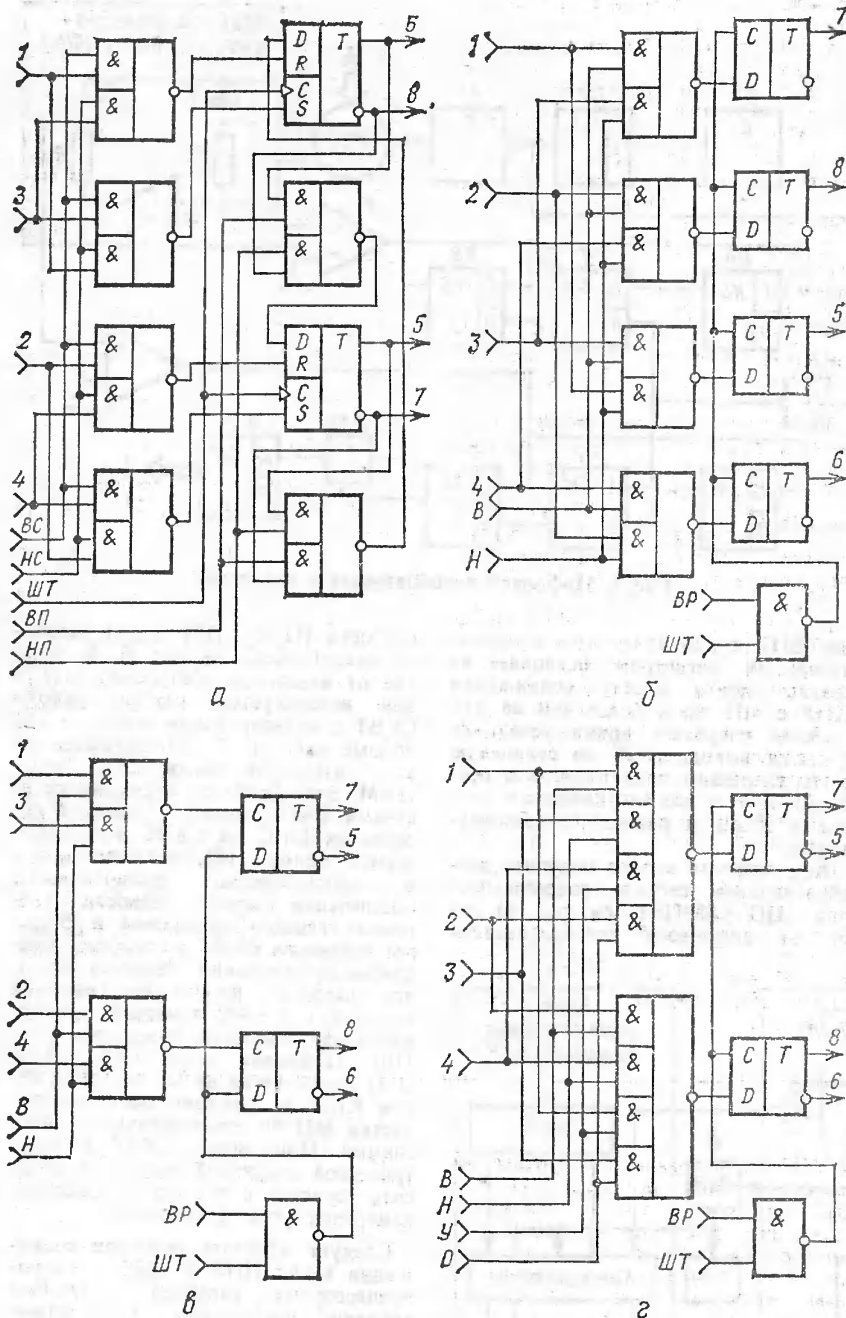


Рис. 6. Распределители полирежимных устройств управления ШЭД:
 ВС, НС, ВП, НП — шины направлений самокоммутации и пошагового управления
 (остальные обозначения из рис. 3 и по тексту)

зовать БИС программируемого контроллера прямого доступа к памяти КР58СВТ57 [20].

Разработка быстродействующих цифровых преобразователей параметров движения и устройств их сопряжения с МП в значительной мере способствовали построению высокоэффективных СМ с ШЭД, работающих в режиме самокоммутации, т. е. бесконтактного двигателя постоянного тока (БДПТ). При работе такого СМ

непосредственно от ЭВМ высшего уровня возникают высокие требования к ее быстродействию [15]. Введение МП, обеспечивающего хранение и обработку информации, поступающей от ЭВМ высшего уровня, позволяет преодолеть указанную трудность. Облегчается задача выбора оптимального алгоритма управления ШЭД для различных алгоритмов функционирования СМ.

Микропроцессор в системе осуще-

ствяет цифровую коррекцию и изменяет алгоритмы и способы управления ШЭД для обеспечения показателей переходных процессов, близких к оптимальным по критерию минимума времени отработки. Формирование закона управления осуществляется на основе анализа текущих координат движения, за которые принята позиционная ошибка и скорость движения, с последующим использованием этой информации для поочередного подключения одного из алгоритмов управления ШЭД [21]. Такое использование МП является более полным, чем построение [22], предусматривающее наличие отдельного регулятора, реализованного аппаратным способом.

Однако следует отметить, что вариант формирования управляющих сигналов для ШЭД программным методом самой микроЭВМ или ПИ (см. рис. 1, б) при самокоммутации не является оптимальным. Эффективность работы ШЭД в этом режиме определяется возможностями распределителя импульсов. Введение в его состав дополнительных логических устройств и элементов памяти расширяет возможности управления ШЭД; обеспечивается управление им как в режиме пошагового управления, так и в самокоммутации. Устройство управления становится полирежимным (ПУУ) [23].

С точки зрения удобства сопряжения с МП предпочтительны распределители ПУУ (рис. 6) [24] на микросхемах серии К155. Они представляют собой сочетание D-триггера, выполняющего функции памяти, и элемента И—ИЛИ—НЕ, осуществляющего реверс. Наиболее общим признаком, характеризующим принцип выполнения таких распределителей, является задание алгоритма управления (типа коммутации) воздействием на входные цепи ПУУ. При ручном управлении процессом применяется схема (рис. 6, а), реализующая автономное реверсирование при неизменном алгоритме для пошагового управления и самокоммутации.

Переход в распределителях (рис. 6, б—г) на D-триггеры (например, К155ТМ7), тактируемые уровнем сигнала, приводит не только к упрощению, но и к расширению возможностей ПУУ. В схеме (рис. 6, г) предусмотрено ступенчатое изменение скорости ШЭД в режиме самокоммутации. Регулируется скорость ШЭД как переключением угла коммутации обмоток управления, так и изменением напряжения на них широкоимпульсной модуляцией (ШИМ). Применение ШЭД с переменным углом коммутации оказывается более эффективным в области высоких частот вращения, чем использование классического БДПТ, имеющего нейтральный угол коммутации. Введение ШИМ способствует повышению плавности движения и устранению

его ступенчатого характера на низких скоростях. Таким образом, микропроцессорное управление ШЭД улучшает качество движения во всем диапазоне скоростей [7].

При программируемом разгоне можно повысить добротность СМ за счет роста скорости отработки при неизменной разрешающей способности, определяющей позиционную точность. Обратная картина имеет место при миншаговом управлении. Перевод ШЭД в режим самокоммутации с ШИМ является более эффективным для повышения добротности как за счет увеличения скорости отработки, так и одновременного роста разрешающей способности двигателя, который работает практически в «гладком» (непрерывном) режиме.

Предложенный вариант построения комплекса МП—СМ (см. рис. 3) обеспечивает решение задачи создания безредукторных СМ высокой точности и надежности на моментных двигателях с дискретной коммутацией и управлением от МП.

ЛИТЕРАТУРА

1. Бесекерский В. А. Проблемы развития систем автоматического управления.— Изв. вузов СССР. Приборостроение, 1982, № 11, с. 20—27.
2. Робототехника/Под ред. Е. П. Попова.— М.: Машиностроение, 1984.—288 с.
3. Платонов А. К. Проблемы разработки микропроцессорных средств для систем управления роботов.— Микропроцессорные средства и системы, 1984, № 1, с. 22—27.
4. Мельник Н. В., Сафонов Ю. М. Электропривод роботов.— М.: ВИНТИ. 1983.—77 с.
5. Доиравчев В. Г., Мейко Б. С. Цифровые преобразователи уг-

ла.— М.: Энергоатомиздат, 1984.—328 с.

6. Хэмден Д. Параллельное управление тремя шаговыми электродвигателями при помощи таймера.— Электроника, 1983, № 20, с. 83—86.

7. Смирнов Ю. С. Микропроцессорные устройства управления сервомеханизмами.— В кн.: Материалы ВК «МП системы», Челябинск, 1984, с. 156, 157.

8. Пэтл В. Л. Интерфейс позволяет микропроцессору управлять шаговым двигателем.— Электроника, 1982, № 15, с. 64, 65.

9. Торгов Ю. И. Учебный центр.— Микропроцессорные средства и системы, 1984, № 1 и 2, с. 77—84; 83—85.

10. Уорд Р. Аппаратный интерфейс и программы для управления шаговым двигателем.— Электроника, 1982, № 26, с. 60—62.

11. Зверев А. Е., Максимов В. П., Мясников В. А. Преобразователи угловых перемещений в цифровой код.— Л.: Энергия 1974.—184 с.

12. Кудряшов Б. А., Макаров В. В., Смирнов Ю. С. и Шишков А. Б. Функциональный преобразователь «угол — амплитуда — код». — Метрология, 1983, № 8, с. 20—26.

13. Лэнтон С. Гибридный преобразователь сельсия-код с большими интегральными схемами.— Электроника, 1981, № 13, с. 43—48.

14. Кудряшов Б. А., Смирнов Ю. С., Шишков А. Б. Амплитудный преобразователь «угол — код» с СКВТ.— Измерительная техника, 1984, № 8, с. 20—21.

15. Смирнов Ю. С. Особенности динамики ШЭД, охваченного местной обратной связью.— Изв. вузов СССР. Приборостроение, 1971, № 6, с. 48—53.

16. Богданов В. Д., Кудряшов Б. А., Смирнов Ю. С. Преобразователь угла в код с переменной структурой.— ПСУ, 1985, № 1, с. 21—23.

17. Смирнов Ю. С. Преобразователи параметров движения в код для САУ с микро-ЭВМ.— В кн.: Материалы ВК «МП системы», Челябинск, 1984, с. 238, 239.

18. Каплун Г. И., Кузнецов В. Р., Пермьяков А. П., Поляшук С. Л. Системы управления электропривода работа на базе микро-ЭВМ. Там же, с. 158, 159.

19. Кудряшов Б. А., Пашук С. П., Смирнов Ю. С. Преобразователь «скорость — код» с СКВТ.— Измерительная техника, 1984, № 10, с. 9—11.

20. Абдуллаев Н. Т., Измайлова Л. З., Тургиев Э. А. Организация работы АЦП в микропроцессорной системе.— ПСУ, 1984, № 10, с. 18—20.

21. Смирнов Ю. С. Цифровой СМ для робототехнических систем с микроЭВМ.— В кн.: Материалы III ВНТК «Робототехнические системы», Челябинск, 1983, ч. 2, с. 78, 79.

22. Воронин С. Г. Электропривод промышленных роботов с управлением от микроЭВМ.— Там же, с. 76, 77.

23. Смирнов Ю. С. О построении многорежимных устройств управления ШЭД.— ЭТ в А/Под ред. Ю. И. Конева.— М.: Сов. радио, 1976, вып. 8, с. 165—169.

24. Смирнов Ю. С., Кудряшов Б. А. Использование интегральных триггеров в коммутаторах БЭД.— В кн.: Обмен опытом в радио-промышленности, 1981, вып. 10, с. 61—63.

Статья поступила
18 сентября 1984 г.

РЖ ВИНТИ, АВТ-85

9Б410. МикроЭВМ для управления станками с числовым программным управлением. 1985, (нем.)

Приводится описание микроЭВМ модели МА производства Швейцарии, используемой для управления систем числового программного управления. Основой микроЭВМ является плата 16-разрядного центрального процессора на основе МП типа 8002, включающая в себя ОЗУ емкостью 8К×16, ПЗУ емкостью 16К×16, БИС параллельного ввода-вывода на 24 порта, БИС счетчиков, параллельных портов (20 каналов) и управления прерываниями, системные часы, индикатор ошибок (на 7-сегментных индикаторах) и стыки с шинами типов УМЕ и С64.

МикроЭВМ укомплектована периферийными платами, имеющими ЦАП

и АЦП на 4 канала с разрешающей способностью 12 бит; цифровые входы счетчиков (3 канала) для преобразования поступающих последовательно импульсных сигналов в параллельную форму; мощные выходные схемы для непосредственного управления и преобразования уровней.

РЖ ВИНТИ, АВТ-85

8.81.12К. Кибернетика и право. Материалы Методологического семинара кафедры советского государственного строительства и права. Академия общественных наук при ЦК КПСС. М., 1984, 130 с. (рус.)

Сборник содержит материалы методологического семинара кафедры

советского государственного строительства и права АОН при ЦК КПСС. В свете решений XXVI съезда КПСС, февральского и апрельского (1984 г.) Пленумов ЦК партии о совершенствовании развитого социализма, соединении социалистической системы хозяйствования с новейшими достижениями научно-технической революции освещаются актуальные проблемы государственно-правового строительства, в частности, создания АСУ в народном хозяйстве, в системе правоохранительных органов (юстиции, прокуратуры, МВД СССР).

Значительное внимание уделено вопросам развития правовой (юридической) кибернетики, перспективам использования разнообразных современных средств кибернетики в целях оптимизации деятельности государственных органов.

УДК 681.325.5—181.4

Н. С. Бамбуров, А. В. Екимов, Ю. В. Ермолин, П. В. Мамаков, А. Н. Сытин

ПРОГРАММИРУЕМЫЙ ГРУППОВОЙ КОНТРОЛЛЕР НА БАЗЕ МКК БИС СЕРИИ К580

Увеличение числа пользователей высокопроизводительными ЭВМ коллективного пользования с развитым программным обеспечением и большим набором внешних устройств ограничивается, как правило, не вычислительными мощностями ЭВМ, а числом штатных интерфейсов асинхронных линий последовательной связи в ЭВМ. При этом с ростом числа удаленных пользователей увеличивается число и общая протяженность кабельных линий связи. Эту проблему можно решить с помощью группового контроллера (ГК), выполняющего роль мультиплексора для группы пользователей [1—3], при этом число кабельных линий связи практически не возрастает.

Групповые контроллеры могут быть выполнены как на основе аппаратной логики с жестким алгоритмом управления, так и программируемыми. Вторые имеют ряд очевидных преимуществ, как и любые программируемые устройства (гибкая перестройка на новый режим работы без изменения аппаратной части), хотя и имеют ограничения по скорости (что несущественно в ряде применений).

Реализации ГК на базе мини- и микроЭВМ, в частности на базе микроЭВМ «Электроника 60» [4, 5], не лишены некоторых недостатков: небольшое число коммутируемых каналов, увеличение которого ограничено размерами и числом плат, которые могут быть подключены к магистрали; сравнительно большая потребляемая мощность и объем аппаратной части, что снижает надежность.

Групповой контроллер (ГК) на основе микропроцессорного комплекта серии К580, соз-

данный в ИФВЭ, в значительной степени лишен указанных недостатков (рис. 1).

Аппаратная часть группового контроллера

Групповой контроллер (рис. 2) выполнен в виде автономного устройства со встроенным источником питания.

МикроЭВМ с интерфейсами асинхронных линий связи выполнена на одной печатной плате. Архитектура ее включает в себя:

процессор — центральный процессорный элемент (ЦПЭ), тактовый генератор, системный контроллер, буферные регистры адресов (БРА); запоминающее устройство (8К байт), выполненное на ПЗУ (2К байт), а также ОЗУ, совместимом по назначению выводов и емкости (2К байт). Пользователь сам выбирает конфигурацию памяти в зависимости от программы;

внешние устройства — 9 каналов последовательного ввода-вывода. Входные и выходные каскады всех каналов имеют оптронные развязки и могут быть как активными, так и пассивными, обеспечивая стандартную «токовую петлю 20 мА»;

генератор скорости передачи обеспечивает скорость от 75 до 9600 бод;

контроллер прерываний;

кнопочный регистр, установленный на передней панели, с которого задаются различные режимы работы ГК: мониторинг (кнопка МОН); мультиплексный (МЛТ), управляющий (УПР).

Встроенный источник питания обеспечивает напряжение ± 5 В и $+12$ В. ГК выполнен в конструктиве модуля системы СУММА [7] шириной 100 мм, может быть вставлен в каркас СУММА и в этом случае использовать напряжение питания от канала каркаса.

МикроЭВМ управляет потоками информации через групповой контроллер согласно программе, хранящейся в ППЗУ. Программа имеет три режима работы, задаваемые переключателем регистра режимов на передней панели, который опрашивается при инициализации микроЭВМ. ОЗУ служит для промежуточной

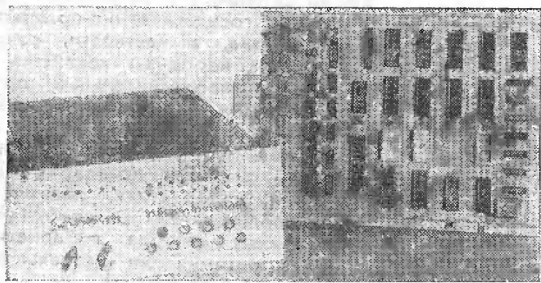


Рис. 1. Общий вид группового контроллера (ГК)

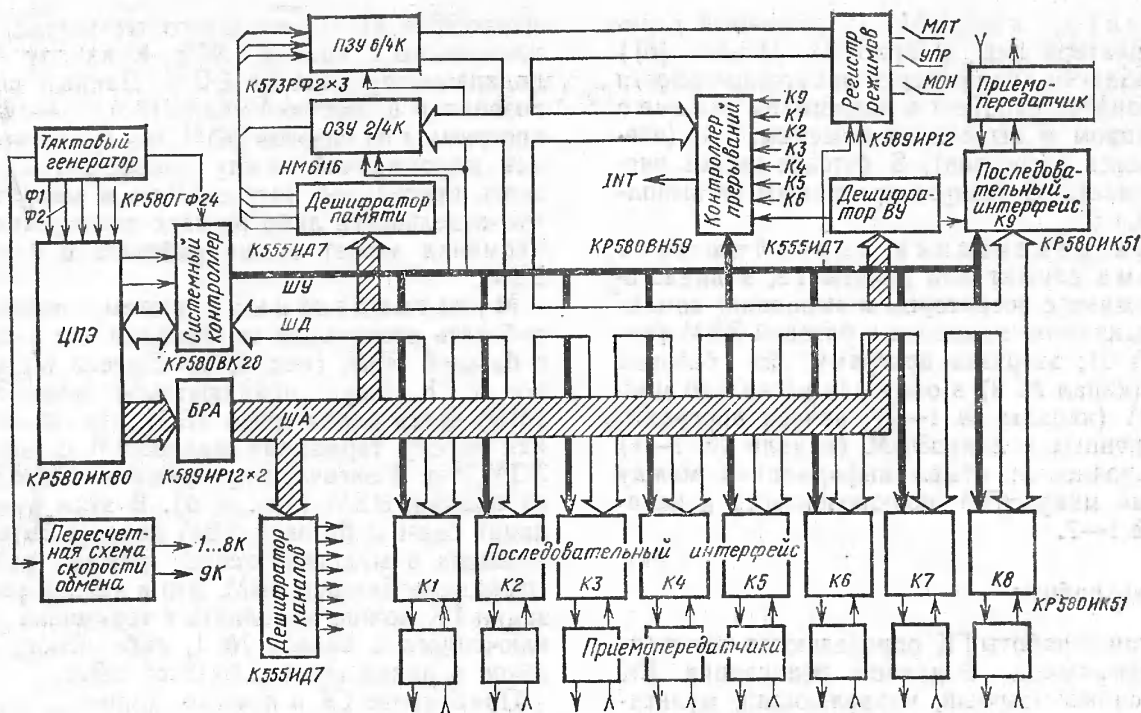


Рис. 2. Структурная схема ГК:

ЦПЭ — центральный процессорный элемент; БРА — буферные регистры адресов; МЛТ — мультиплексный; УПР — управляющий; МОН — мониторный

буферизации. При поступлении информации в какой-либо из каналов ввода-вывода от этого канала вырабатывается прерывание микроЭВМ, запускающее программу пересылки информации. Контроллер прерываний позволяет приоритетно кодировать запросы от различных каналов, а также маскировать отдельные прерывания, что необходимо для «отключения» тех каналов, к которым не подключены внешние устройства.

Из 9 каналов ввода-вывода один служит для связи с базовой ЭВМ, а остальные 8 — для подключения терминалов либо микроЭВМ. Конфигурация определяется только программным обеспечением, используемым в ГК.

Программное обеспечение

Программное обеспечение (ПО) ГК включает в себя: стандартное ПО базовой ЭВМ, позволяющее работать линии в мультиплексном режиме с несколькими заданиями (каждый передаваемый символ сопровождается управляющим, указывающим номер задания), и специализированное ПО (4К байт) группового контроллера (определяется задачами, выполняемыми ГК), реализованное на языке ассемблера для микропроцессора серии К580. Специализированное ПО состоит из пяти программных блоков: таблицы прерываний, блока инициализации и реализации мультип-

лексного режима, монитора микроЭВМ, блока реализации управляющего режима.

В таблице прерываний для каждого уровня прерывания определен адрес перехода на соответствующую программу обработки.

Уровень прерывания 0 (наивысший)	Общий сброс ГК
1	от канала № 9
2	от канала № 1
3	от канала № 2
4	от канала № 3
5	от канала № 4
6	от канала № 5
7	от канала № 6

В блоке инициализации программы инициализируются каналы ввода-вывода, контроллер прерываний и определяется режим работы ГК.

Алгоритм работы блока реализации мультиплексного режима следующий. По приходе информации от какого-либо канала происходит прерывание. Программа прерывания считывает эту информацию и пересылает в соответствующий канал. При этом, если информационный символ (байт) передается от пользователя (каналы 1—6) в базовую ЭВМ (канал № 9), то к нему добавляется «байт сопровождения», определяющий номер канала пользователя. Из каждых двух байтов, приходящих из базовой ЭВМ («байт сопровождения» + информационный байт), выделяется второй и отсылается одному из пользователей.

Монитор микроЭВМ (упрощенный вариант монитора для микроЭВМ МЭ-80 [6]) предназначен для проверки работоспособности ГК. Монитор работает в диалоговом режиме с оператором и выполняет команды: X (чтение-запись регистров), S (чтение-запись ячеек памяти), G (запуск программы на выполнение).

Блок реализации управляющего режима служит для работы ГК в диалоговом режиме с оператором и выполняет команды: подключение дисплея к базовой ЭВМ (канал № 9); загрузка программ из базовой ЭВМ (канал № 9) в одну или несколько микроЭВМ (каналы № 1—7); запуск программ, загруженных в микроЭВМ (каналы № 1—7) на выполнение; обмен информацией между любыми микроЭВМ, подключенными к каналам № 1—7.

Режимы работы

Режимы работы ГК определяются записанной программой. В данной реализации ГК имеются мониторный, управляющий, мультиплексный режимы.

Мониторный режим предназначен для отладки микроЭВМ группового контроллера. Дисплей подключается к каналу № 8, и микроЭВМ работает под управлением программы монитора [6].

Управляющий режим (рис. 3) обеспечивает возможность управления несколькими

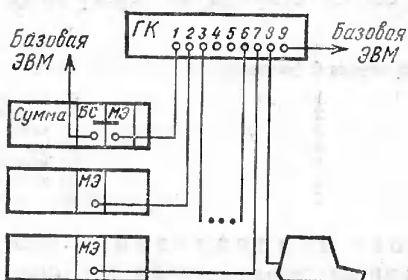


Рис. 3. Управляющий режим ГК

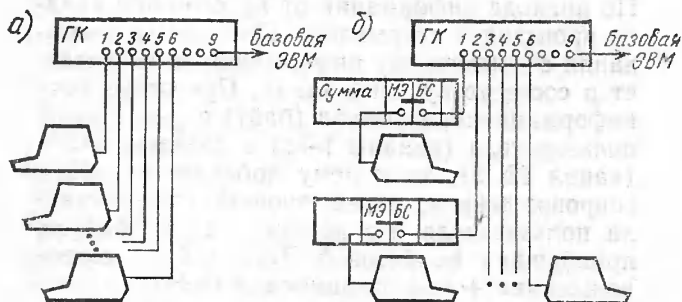


Рис. 4. Мультиплексный режим ГК

микроЭВМ (1—7) от одного терминала, подключенного к каналу № 8. К каналу № 9 подключается базовая ЭВМ. Данный режим позволяет в любую микроЭВМ загружать программы из базовой ЭВМ, осуществлять обмен информацией между микроЭВМ и запускать программы, загруженные в микроЭВМ (по отдельности либо во всех одновременно). Терминал может также работать с базовой ЭВМ.

Мультиплексный режим позволяет работать нескольким терминалам (от 1 до 6) с базовой ЭВМ (рис. 4, а). Вместо терминалов к ГК могут подключаться микроЭВМ (МЭ) через блоки связи (БС), что обеспечит как работу терминала микроЭВМ с базовой ЭВМ, так и загрузку программ в микроЭВМ из базовой ЭВМ (рис. 4, б). В этом режиме линия связи с базовой ЭВМ должна быть переведена в мультиплексный режим работы командами базовой ЭВМ. Это в данной реализации ГК можно выполнить с терминала, подключенного к каналу № 1, либо непосредственно к линии связи с базовой ЭВМ.

Применение ГК в качестве узлового коммутационного процессора — один из возможных вариантов обеспечения межмашинного обмена для построения многомашинных комплексов. В этом случае число линий связи между ЭВМ заметно уменьшается при числе ЭВМ больше трех (рис. 5).

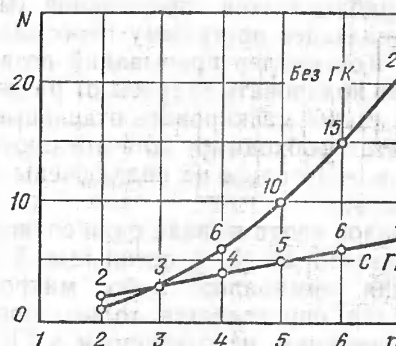


Рис. 5. Число линий связи между ЭВМ при различной организации межмашинного обмена; n — число ЭВМ; N — число линий связи.

Недостатки данного решения вытекают из структуры типа «звезда», но при малой интенсивности обмена и надежной аппаратуре они не существенны. Эти условия выполняются при построении многомашинных комплексов на основе микроЭВМ, где они работают, как правило, параллельно и независимо, и обмен данными между ними невелик, но при этом требуется контроль над всем комплексом со стороны главной ЭВМ.

Адрес для справок: 142284, г. Серпухов, Моск. обл., Институт физики высоких энергий.

А. М. Петух, А. Н. Романюк, О. А. Подольский

ПРЕОБРАЗОВАТЕЛЬ ИНТЕРФЕЙСОВ ИРПС — „ОБЩАЯ ШИНА”

При обслуживании вычислительных комплексов часто необходимо подключать дополнительные функциональные средства, ориентированные на определенные области применения и специальные режимы работы. Для оперативного подключения периферийных устройств, имеющих систем-

ные характеристики, различные с каналом ЭВМ, т. е. для дополнения моделирующих интерфейсов комплекса и обеспечения требуемого режима передачи целесообразно ввести простые функциональные узлы.

Для оперативного обмена инфор-

мацией человека с центром обработки данных в комплексах СМ ЭВМ в качестве пульта оператора или пульта программиста используются алфавитно-цифровые видеотерминалы ВТА 2000—2, ВТА 2000—3. Обмен информацией между ними и процессором обеспечивает интерфейс для радиального подключения устройств с параллельной передачей информации (ИРПР). Для перехода с интерфейса ИРПР на интерфейс «Общая шина» используются типовые элементы замены (ТЭЗ) БЭ002 и БЭ810М, которые устанавливаются в процессоре мини-ЭВМ СМ-3, СМ-4. Интерфейсный кабель (6 м), соединяющий процессор и видеотерминал, не позволяет работать с удаленным аб-

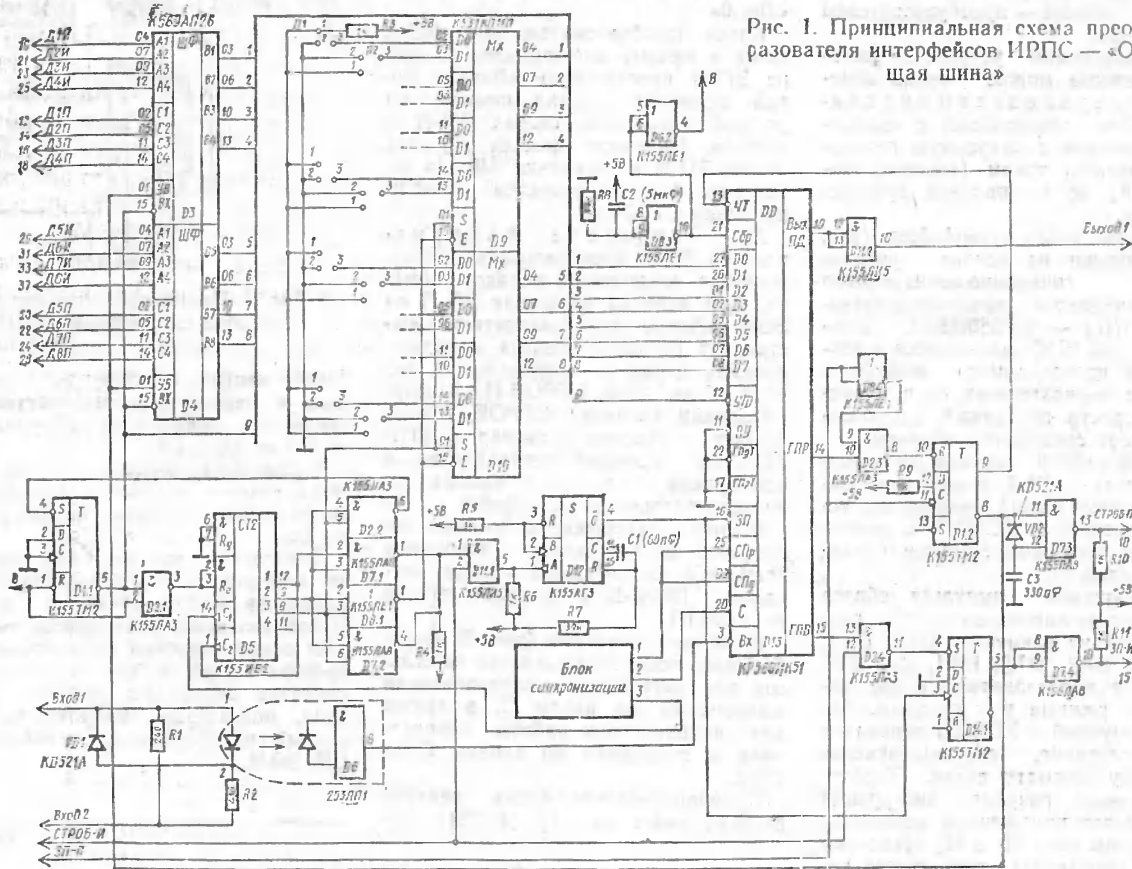


Рис. 1. Принципиальная схема преобразователя интерфейсов ИРПС — «Общая шина»

ЛИТЕРАТУРА

1. Зайчкин Н. И., Лысунков Э. М., Рябов А. А. — Электронная техника. Сер. 9. «Экономика и системы управления», 1982, вып. 4 (45), с. 47—50.
2. DEC System 10. System Reference Manual. — DEC, Maunard, Massachusetts, USA, 1976.
3. Polandri E. M., Struck-Zimmermann H. W. A Multi-Host Front End Concentrator System for Asynchronous Consoles. — CERN DD/74/14.
4. Куянов Ю. В., Петухов В. А., Савин Н. П. и др. — Препринт 83—107. — Серпухов: ИФВЭ, 1983.
5. Кузьменко В. Г., Лукьянцев А. Ф., Мак-

симов Г. М. и др. — Препринт 83—135. — Серпухов: ИФВЭ, 1983.

6. Говорун В. Н., Екимов А. В., Соколов А. П. Программное обеспечение микропроцессорных систем на базе микропроцессора КР5801К80. — Препринт 82—194. — Серпухов: ИФВЭ, 1982.

7. Алферова О. И., Бушнин Ю. Б., Денисенко А. А. и др. Система унифицированных модулей многоканального анализа СУММА. Основные характеристики, источник питания. — Препринт 74—122. — Серпухов: ИФВЭ, 1974; ПТЭ, 1975, № 4, 56.

Статья поступила 24 января 1985 г.

нентом. В большинстве же современных систем управления необходимо удалять некоторые периферийные устройства на большие расстояния, а также объединять несколько ЭВМ, удаленных друг от друга.

Преобразователь интерфейсов обеспечивает протокол обмена между устройствами с интерфейсом радиальным последовательным (ИРПС) и ИРПР, а работая с ТЭЗами БЭ002, БЭ810М, осуществляет переход ИРПС—«Общая шина». В этом случае функции адресного селектирования и организации режима прерывания выполняются ТЭЗами, а процесс преобразования последовательных посылок в параллельный код и организации асинхронного обмена — преобразователем интерфейсов.

В предлагаемом устройстве параметры режима можно гибко изменять перепрограммированием. Оно осуществляет с удаленным абонентом асинхронную передачу постоянным током (токовая петля 20 мА) по 4-проводной дуплексной связи.

Преобразователь интерфейсов (рис. 1) реализован на основе универсального синхронно-асинхронного программируемого приемопередатчика (УСАПП) — КР5801К51. Универсальность БИС заключается в возможности программного изменения форматов передаваемых слов, режима и скорости передачи*. Устройство работает следующим образом.

При включении питания на выходе элемента D8.3 появляется импульс положительной полярности, устанавливающий УСАПП в режим ожидания последовательности управляющих слов.

По окончании импульса сброса блок программирования БИС, реализованный на элементах D1.1, D2.1, D2.2, D5, D7.1, D7.2, D8.1, D9, D10, формирует последовательно две инструкции: режима и команды. Запись инструкций в УСАПП переводит его в состояние, соответствующее желаемому формату связи. Требуемые значения разрядов инструкций определяются положением переключателей на наборном поле П₁ и П₂. Наборное поле П₁ определяет инструкцию режима, а П₂ — инструкцию команды. Наличие связи между контактами 1 и 3 соответствует «Лог.1» в соответствующих разрядах управляющих слов, а между контактами 2 и 3 — «Лог.0». Инструкции режима и команды на шину данных УСАПП коммутируются элементами D9, D10.

Состояния двонечно-десятичного счетчика D5 определяют временную последовательность установки управ-

ляющих слов на шину данных БИС и формирования сигналов записи. Соответствие между состояниями счетчика и выполняемыми операциями приведено в таблице.

После записи инструкции команды на выходе ГПД УСАПП устанавливается напряжение «Лог.1», сигнализируя о готовности передатчика БИС принять данные с канала. Этот сигнал совместно с сигналами СТРОБ-И служит для формирования сигнала ЗП-И.

«Лог.1» на линии ЗП-И означает, что приемник не готов к приему нового символа, но принял предыдущий символ, если он был. Приемник не должен принимать данные, пока сигнал СТРОБ-И не перейдет в «Лог.0».

Когда преобразователь интерфейса готов к приему информации, на шине ЗП-И присутствует «Лог.0». Когда приемник принял символ, он должен установить сигнал ЗП-И в «Лог.1». Приемник должен держать линию ЗП-И в состоянии «Лог.1» до тех пор, пока не появится «Лог.1» на линии СТРОБ-И.

При передаче информации в ЭВМ преобразователь обеспечивает следующий порядок обмена. При наличии на шинах ЗП-П от ЭВМ «Лог.0» преобразователь выставляет на шине данных информационное слово и сопровождает его «Лог.0» на шине СТРОБ-П. В формировании сигнала СТРОБ-П принимает участие сигнал ГПР УСАПП, который устанавливается при приеме микросхемой данных по последовательному интерфейсу.

Элемент задержки, собранный на элементе D7.3, создает требуемое временное соотношение между сигналом СТРОБ-П и сигналом чтения из УСАПП.

Блок синхронизации (рис. 2) вырабатывает последовательность импульсов для внутренней синхронизации микросхемы по входу С, а также для синхронизации работы передатчика и приемника по входам СПД, СПР.

Приемопередатчик блока синхронизации делит частоту (1:16); тре-

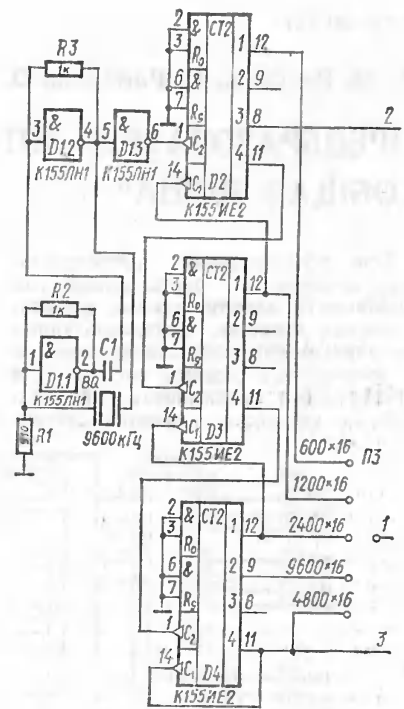


Рис. 2. Принципиальная схема блока синхронизации

буемая частота синхронизации приемника и передатчика выбирается установкой переключки на наборном поле ПЗ из D1...D4.

Чтобы электрическое заземление гальванически разделить, используется оптронная развязка на микросхеме D6.

Длительная эксплуатация показала надежность преобразователя интерфейсов ИРПС—«Общая шина». Преобразователь интерфейсов выполняет дополнительные функциональные преобразования и дает возможность изменять параметры режимов передачи, оперативно подключать различные периферийные устройства к СМ ЭВМ.

Таблица

Состояние счетчика				Выполняемая операция
2 ⁰	2 ¹	2 ²	2 ³	
0	1	0	0	Установка инструкции режима
1	1	0	0	Запись инструкции режима
0	0	1	0	Установка инструкции команды
1	0	1	0	Запись инструкции команды
0	1	1	0	Конец программирования

РЖ ВИНТИ, АВТ-85

8.81.594. Универсальный цифровой программируемый автомат на базе микроЭВМ СМ-1800 для логического и непрерывно-дискретного автоматического управления оборудованием и технологическими процессами на предприятиях промышленности.

Изложены принципы разработки базового программного обеспечения для микроЭВМ типа СМ-1800 — универсальный цифровой программируемый автомат (УЦПА) для имитации релейно-контактной логики на ЭВМ. Рассмотрены назначение УЦПА, базового программного обеспечения и варианты комплектации УЦПА.

* Алексеев А. Г., Галицын А. А., Ивашников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. — М.: Радио и связь, 1984.

УДК 681.3:621.316.71

С. А. Михайлов

СИСТЕМА УПРАВЛЕНИЯ ЭЛЕКТРОПРИВОДАМИ СУДОВЫХ ГРУЗОВЫХ КОМПЛЕКСОВ

Хорошо известно, что лучшей источник энергии — ее экономия. Внедрение энергосберегающей техники и технологии особенно актуально для автономных энергетических систем с ограниченными ресурсами и возможностями их пополнения. Пример — судовые грузовые комплексы (СГК) [1, 2], получающие энергию от судовой электростанции (СЭС). В судовых грузоподъемных механизмах (ГПМ) для достижения широкого диапазона изменения скорости перемещения груза наиболее часто применяют электрический привод (ЭП) — многоскоростные АД, т. е. асинхронные двигатели [3]. Производительность и грузоподъемность ГПМ во многом зависят от мощности его электроприводов, поэтому тенденция к росту мощности электродвигателей устойчива.

В настоящее время типична следующая комплектация электроэнергетической установки грузового судна [6]: СЭС — четыре—шесть параллельно работающих дизель-генераторов с мощностью каждого синхронного генератора (СГ) 400... 800 кВт и более, асинхронные по-

люсопереключаемые электродвигатели грузовых устройств мощностью 60...90 кВт и более каждый. Число ГПМ на одном судне — 16...22. Во время стоянки судна для грузовых операций ЭП ГПМ — основные потребители СЭС, а мощность их электродвигателей соизмерима с мощностью генераторов электростанции (см. фото на 4-й стр. вкладки).

При одновременном включении АД возникают глубокие провалы величины (до 30...50 %) и частоты (до 5...20 %) напряжения судовой сети, отрицательно влияющие на работу других потребителей. Провалы максимальны при включении высших скоростей АД, так как мощность ЭП при этом наибольшая.

Для снижения провалов в СЭС обычно включают на параллельную работу несколько генераторных агрегатов, чтобы обеспечить необходимый запас мощности. Многим судоводителям и электромеханикам знакома ситуация, когда при грузовых операциях работают два-три синхронных генератора, загруженных в установившемся режиме едва напо-

ловину, хотя и один из них полностью при загрузке до 80 % обеспечил бы потребность судна в электроэнергии. Таким образом, нужно рационально строить структуру энергетической сети с ГК и алгоритмы оптимального управления его технологическими режимами.

Известные пути решения этих проблем [3, 6, 10] позволяют сделать вывод, что эффективным техническим средством их реализации могут служить системы управления на основе БИС [1, 5, 8] с широкими функциональными возможностями. Они компактны и надежны, способны реализовать требуемые законы регулирования. Представление информации в виде цифрового кода упрощает согласование с внешними ЭВМ [4, 7, 8].

Структура системы изображена на рис. 1. Многоагрегатная (дизель Д — синхронный генератор СГ) судовая электростанция со своей системой управления (СУ СЭС) питает через главный распределительный щит (ГРЩ) потребителей электроприводы ГПМ грузового комплекса. Шина питания к ЭП подключается силовым автоматическим выключателем А, а управление осуществляется через бесконтактные тиристорные коммутаторы (ТК). Нижний уровень системы — автоматы локального управления (АЛУ) с постоянными запоминающими устройствами (ПЗУ). Органом управления конкретным ГПМ служит традиционный, а с точки зрения эргономики и перспективный командоаппарат локального управления (КЛУ). Пульт диспетчерского управления (ПДУ) задает исходную информацию общего характера: сколько ГПМ будет работать и какие, сколько СГ будут работать и какие, какие потребители ответственные и т. д.

Электростанция, грузовой комплекс и устройства управления объединены в единую систему. Стратегию управления определяет микроЭВМ (верхний уровень). Алгоритм работы микроЭВМ предусматривает установку для каждого потребителя его приоритета с учетом требуемого быстродействия и важности механизма.

При запоминании очередности поступления запросов на включение запуска их электроприводов разде-

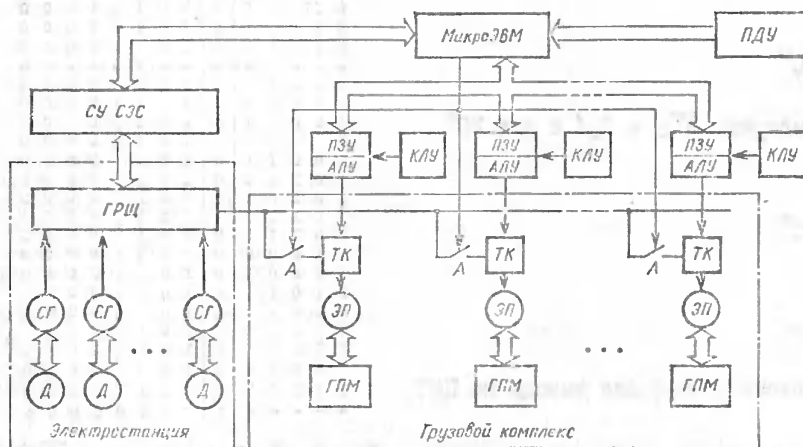


Рис. 1. Автономная энергетическая система

ляются во времени с учетом установленных приоритетов и имеющегося резерва мощности. При несоответствии требуемой и включенной мощности в СЭС параллельно подключается очередной дизель-генератор. Длительность провалов напряжения при пуске мощных электродвигателей и требуемая при этом повышенная мощность источника электроэнергии составляют единицы секунд [6]. Задержка запуска на это время высшей скорости электропривода другого ГПМ практически не скажется на изменении произво-

дительности грузового комплекса, но даст выигрыш в требуемом резерве мощности СЭС и соответственно в расходе горюче-смазочных материалов и моторесурса генераторных агрегатов.

В силовых цепях ЭП необходимо применение устройств защиты с зависимой от уровня тока выдержкой времени на отключение, которые должны срабатывать не только при коротких замыканиях, но и при перегрузках электродвигателей.

Согласно рекомендациям МЭК 255—4 [9], время-токовая характе-

Оператор	Комментарий
1 СЕЛЕКТ<12, 32, 0>	Программирование интерфейса ввода с АЦП
2 ВВОД М<,,,СА00, 7>	Ввод и обозначение входных переменных
3 М00	$\left. \begin{matrix} K_1^B \\ K_1^H \\ K_2^B \\ K_2^H \\ K_3^B \\ K_3^H \end{matrix} \right\}$ Анализ входных переменных
4 СА01="1"ПЕРЕХ М01	
5 СА04="1"ПЕРЕХ М02	
6 СА02="1"ПЕРЕХ М03	
7 СА05="1"ПЕРЕХ М04	
8 СА03="1"ПЕРЕХ М05	
9 СА06="1"ПЕРЕХ М06	
10 ПЕРЕХ М00	
11 М01	
12 М09	
13 СА01=>САИ01	$\left. \begin{matrix} IC \\ TB \end{matrix} \right\}$ IC ^B
14 СА01=>САИ00	
15 СА01=>САИ04	
16 0.1=>А00	
17 ЦИКЛ<А00, 0.1, 3>	$\left. \begin{matrix} IC \\ TH \end{matrix} \right\}$ IC ^H
18 ЦИКЛ*	
19 М02	
20 М10	
21 СА04=>САИ01	$\left. \begin{matrix} IC \\ TH \end{matrix} \right\}$ IC ^H
22 СА04=>САИ00	
23 СА04=>САИ05	
24 0.1=>А00	
25 ЦИКЛ<А00, 0.1, 3>	$\left. \begin{matrix} IC \\ TH \end{matrix} \right\}$ IC ^H
26 ЦИКЛ*	
27 М03	
28 М08	
29 А00<1.3ПЕРЕХ М09	$\left. \begin{matrix} 2C \\ TB \end{matrix} \right\}$ 2C ^B
30 СА02=>САИ02	
31 СА02=>САИ00	
32 СА02=>САИ04	
33 0.1=>А00	$\left. \begin{matrix} 2C \\ TH \end{matrix} \right\}$ 2C ^H
34 ЦИКЛ<А00, 0.1, 2.4>	
35 ЦИКЛ*	
36 М04	
37 М07	$\left. \begin{matrix} 2C \\ TH \end{matrix} \right\}$ 2C ^H
38 А00<1.3ПЕРЕХ М10	
39 СА05=>САИ02	
40 СА05=>САИ00	
41 СА05=>САИ05	$\left. \begin{matrix} 2C \\ TH \end{matrix} \right\}$ 2C ^H
42 0.1=>А00	
43 ЦИКЛ<А00, 0.1, 2.4>	
44 ЦИКЛ*	
45 М05	$\left. \begin{matrix} 3C \\ TB \end{matrix} \right\}$ 3C ^B
46 А00<2.4ПЕРЕХ М06	
47 СА03=>САИ03	
48 СА03=>САИ00	
49 СА03=>САИ04	$\left. \begin{matrix} 3C \\ TH \end{matrix} \right\}$ 3C ^H
50 М06	
51 А00<2.4ПЕРЕХ М07	
52 СА06=>САИ03	
53 СА06=>САИ00	$\left. \begin{matrix} 3C \\ TH \end{matrix} \right\}$ 3C ^H
54 СА06=>САИ05	
55 СЕЛЕКТ<13, 36, 0>	
56 ВЫВОД М<,,,САИ00, 7>	

Рис. 2. Фрагмент программы процессора «Искра 1256» для управления трехскоростным реверсивным электроприводом

Адрес					Содержимое								
A ₀	A ₁	A ₂	A ₃	A ₄	R	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	1	0	1	0	0	0	0
0	0	0	1	0	0	1	1	0	0	1	0	0	0
0	0	0	1	1	0	1	1	0	1	0	0	0	0
0	0	1	0	0	0	1	1	0	0	0	1	0	0
0	0	1	0	1	0	1	1	0	1	0	0	0	0
0	0	1	1	0	0	1	1	0	1	0	0	0	0
0	0	1	1	1	0	1	1	0	1	0	0	0	0
0	1	0	0	0	0	1	1	0	0	0	0	1	0
0	1	0	0	1	0	1	1	0	0	0	0	0	0
0	1	0	1	0	0	1	1	0	0	1	0	0	0
0	1	0	1	1	0	1	1	0	1	0	0	0	0
0	1	1	0	0	0	1	1	0	0	1	0	0	0
0	1	1	0	1	0	1	1	0	0	1	0	0	0
0	1	1	1	0	0	1	1	0	0	1	0	0	0
0	1	1	1	1	0	1	1	0	0	1	0	0	0
1	0	0	0	0	0	1	1	0	0	0	0	0	1
1	0	0	0	1	0	1	1	0	1	0	0	0	0
1	0	0	1	0	0	1	1	0	1	0	0	0	0
1	0	0	1	1	0	1	1	0	1	0	0	0	0
1	0	1	0	0	0	1	1	0	0	0	1	0	0
1	0	1	0	1	0	1	1	0	0	1	0	0	0
1	0	1	1	0	0	1	1	0	0	1	0	0	0
1	0	1	1	1	0	1	1	0	0	1	0	0	0
1	1	0	0	0	0	1	1	0	0	0	1	0	0
1	1	0	0	1	0	1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1	0	0	0	1	0	0
1	1	0	1	1	0	1	1	0	0	0	1	0	0
1	1	1	0	0	0	1	1	0	0	0	1	0	0
1	1	1	0	1	0	1	1	0	0	0	1	0	0
1	1	1	1	0	0	1	1	0	0	0	1	0	0
1	1	1	1	1	0	1	1	0	0	0	1	0	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0

Рис. 3. Программирование ПЗУ D1 локального автомата

ристка устройства защиты в данном случае должна иметь «очень сильно выраженную обратную зависимость». В микропроцессорных системах выдержки времени обычно

Адрес					Содержимое								
A ₀	A ₁	A ₂	A ₃	A ₄	R	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	1	0	0	0	1	0
0	0	0	1	1	0	1	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	1	0	0	1	0	0
0	0	1	0	1	0	1	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0	1	0	0	1	0	0
0	0	1	1	1	0	1	0	1	0	0	1	0	0
0	1	0	0	0	0	1	0	1	0	1	0	0	0
0	1	0	0	1	0	1	0	1	0	1	0	0	0
0	1	0	1	0	0	1	0	1	0	1	0	0	0
0	1	0	1	1	0	1	0	1	0	1	0	0	0
0	1	1	0	0	0	1	0	1	0	1	0	0	0
0	1	1	0	1	0	1	0	1	0	1	0	0	0
0	1	1	1	0	0	1	0	1	0	1	0	0	0
0	1	1	1	1	0	1	0	1	0	1	0	0	0
1	0	0	0	0	0	1	0	1	0	0	0	0	1
1	0	0	0	1	0	1	0	1	0	0	0	0	1
1	0	0	1	0	0	1	0	1	0	0	0	0	1
1	0	0	1	1	0	1	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0	1	0	0	0	0	1
1	0	1	0	1	0	1	0	1	0	0	0	0	1
1	0	1	1	0	0	1	0	1	0	0	0	0	1
1	0	1	1	1	0	1	0	1	0	0	0	0	1
1	1	0	0	0	0	1	0	1	0	0	0	0	1
1	1	0	0	1	0	1	0	1	0	0	0	0	1
1	1	0	1	0	0	1	0	1	0	0	0	0	1
1	1	0	1	1	0	1	0	1	0	0	0	0	1
1	1	1	0	0	0	1	0	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	0	0	0	0	1
1	1	1	1	0	0	1	0	1	0	0	0	0	1
1	1	1	1	1	0	1	0	1	0	0	0	0	1
1	1	1	1	1	1	1	0	0	0	0	0	0	0

Рис. 4. Программирование ПЗУ D2 локального автомата

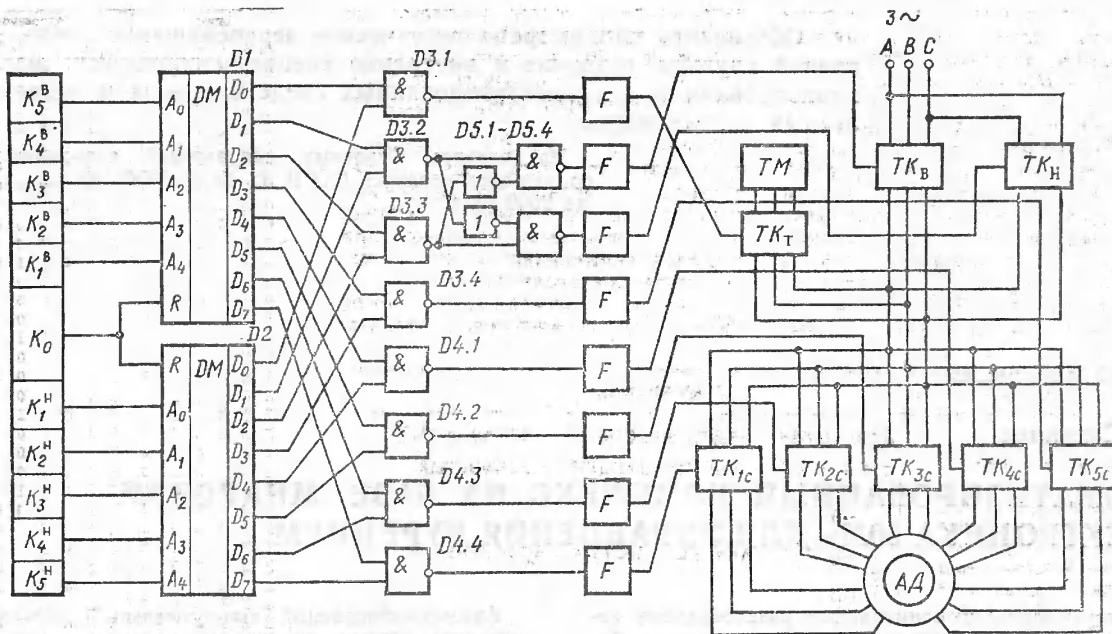


Рис. 5. Автомат локального управления электроприводом грузоподъемного механизма:
D1, D2 — К155РЕЗ, D3, D4, D5 — К155ЛА3

реализуются подсчетом импульсов генератора до заполнения двоичного счетчика. Такая система имеет линейную временную характеристику. Для получения требуемой временной зависимости следует использовать семейство таких линейных характеристик с программным переключением с одной на другую в зависимости от абсолютного значения уровня тока электродвигателя. Переход на различные время-токовые характеристики учитывается в программе микропроцессора. Для этого в ПЗУ записывается внутренняя просмотрная таблица со значениями времени срабатывания защиты при данном токе. После отсчета выбранного времени и отсутствия снижения тока формируется команда на отключение защищаемого ЭП.

На рис. 2 дан фрагмент программы процессора микроЭВМ «Искра 1256», управляющей трехскоростным реверсивным ЭП судового ГПМ. Входные переменные здесь — логические сигналы локального командоаппарата: K^a_1, K^a_2, K^a_3 — включение первой (1С^a), второй (2С^a) и третьей (3С^a) скоростей движения «вперед»; K^b_1, K^b_2, K^b_3 — соответственно «назад»; выходными — сигналы включения соответствующих тиристорных коммутаторов; 1С, 2С, 3С — включение ТК обмоток первой, второй, третьей скоростей; Т — обмотки тормозного электромагнита (ТМ); В и Н — ТК реверсивной группы «вперед» и «назад». При поступлении входного сигнала сразу на включение высшей скорости осуществляется последова-

тельный разгон ЭП с выдержками времени на каждой из промежуточных скоростей.

Объем данного участка программы — 256 байт. Это составляет около 1/60 памяти микропроцессора в 16К. Таким образом, с помощью серийной микроЭВМ возможно локальное управление электроприводами всех ГПМ судна; оперативное управление составом СЭС и приоритетным включением потребителей; диспетчерское управление на системном уровне всем судовым грузовым комплексом.

ПЗУ D1 и D2 автомата локального управления пятискоростным реверсивным АД [12] (рис. 5) запрограммированы в соответствии с рис. 3 и 4. Для односторонности входных и восьми выходных сигналов (см. рис. 2) требуется одно ПЗУ емкостью 16384 бит. Два ПЗУ объединяются так, что сигналы КЛЮ скоростей направления вращения «вперед» ЭП ($K^a_1...K^a_3$) поданы на адресные входы одного, сигналы направления «назад» ($K^b_1...K^b_3$) — на адресные входы другого; сигнал исходного положения K_0 — на входы стробирования обоих ПЗУ (их выходы объединены через схемы ИЛИ D3—D4). В каналах В и Н включен узел блокировки D5. Предложенная конфигурация позволяет использовать ПЗУ емкостью 256 бит, т. е. в 64 раза меньше, чем требовалось бы.

Разработанные автоматы локального управления с тиристорными регуляторами в 1983 г. демонстрировались на ВДНХ СССР.

ЛИТЕРАТУРА

1. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы. — М.: Радио и связь, 1981.
2. Бъчваров Т. Т., Васильев В. Н., Михайлов С. А. Современные алгоритмы управления судовыми грузоподъемными механизмами. — В кн.: Труды Юбилейной научно-технической конференции Высшего машино-электротехнического института. — Варна, 1983.
3. Васильев В. Н., Михайлов С. А., Бъчваров Т. Т. Алгоритмы микропроцессорного управления электроприводами судовых кранов. — Судостроение, 1984. № 3.
4. Зиссос Д. Проектирование систем на микропроцессорах. — Киев: Техника, 1982.
5. Клигман Э. Проектирование микропроцессорных систем. — М.: Мир, 1980.
6. Конечные автоматы судовых энергетических установок / В. Н. Васильев, А. Н. Пипченко, А. А. Виницкий, С. А. Михайлов. — Киев, Вища школа, 1984.
7. Кофрон Дж. Технические средства микропроцессорных систем. — М.: Мир, 1983.
8. Микропроцессорные БИС и микроЭВМ / Под ред. А. А. Васенкова. — М.: Сов. радио, 1980.
9. Паперно Л. Б. Бесконтактные токовые защиты электроустановок. — М.: Энергоиздат, 1983.
10. Радин В. И., Быков Ю. М., Пар И. Т. Микропроцессоры в автомобильных системах электропитания. — Электротехника, 1981, № 2.
11. Уокерли Дж. Архитектура и программирование микроЭВМ. — М.: Мир, 1984, т. 12.
12. А. с. СССР 1050084. Устройство для управления многоскоростным асинхронным электродвигателем / С. А. Михайлов. — Опубл. в Б. И., 1983, № 39.

Статья поступила 20 апреля 1985 г.

Обеспечить планомерное техническое переоснащение геологоразведочной службы, создание и внедрение специализированных высококомпьютеризированных и автоматизированных средств поиска и разведки полезных ископаемых...

Из проекта Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года

УДК 622.24.007:681.322

Г. Н. Семенцов

АВТОМАТИЗИРОВАННЫЙ КОМПЛЕКС НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60М» ДЛЯ УПРАВЛЕНИЯ БУРЕНИЕМ

В промышленности применяются разнообразные автономные системы контроля параметров процесса бурения: ПКБ-2, ПКБ-3, СКУБ, СКД-Полтава, Б-7, АБУБ, «Карпаты» и др. [1—4], рассчитанные в основном на регистрацию небольших потоков информации (7...11 параметров).

Чтобы решить такие сложные [1] задачи, как идентификация параметров математической модели объекта, обнаружение, прогнозирование границ пластов, прогнозирование, обнаружение и распознавание предаварийных ситуаций и осложнений, оперативная оптимизация режимов спуска, бурения и подъема и других, требуется большая скорость обработки информации [5, 6]. Поскольку доля затрат на углубление скважин достигает 60 % от общего объема затрат на бурение, необходима разработка системы оптимального управления процессом углубления скважин, способной обрабатывать большие потоки достоверной информации о технологическом процессе.

Передвижной автоматизированный комплекс, управляющий бурением нефтяных и газовых скважин, построен на базе микроЭВМ «Электроника 60М» в замкнутом контуре управления, работающей совместно с автоматическими системами регулирования осевой нагрузки на долото, частоты вращения долота, расхода бурового раствора — супервизорное управление (см. фото на 1-й стр. вкладки).

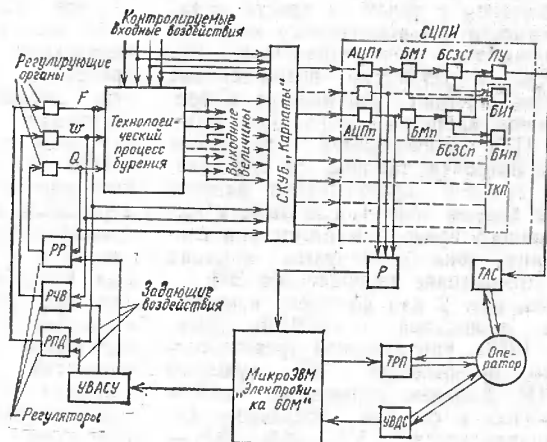
Получив информацию о регулирующих воздействиях, контролируемых входных воздействиях и выходных величинах, микроЭВМ «Электроника 60М» на основе составленных программ обрабатывает эту информацию, идентифицирует параметры математической модели и задает оптимальные воздействия на регуляторы частоты вращения, расхода бурового раствора и подачи долота. Кроме того, микроЭВМ определяет время окончания рейса, границы пластов разрабатываемых пород, зоны с аномально-высокими пластовыми давлениями, ряд предаварийных ситуаций и осложнений. Объем программ равен 8 К байт.

При супервизорном управлении связь с оператором носит информационный характер и микроЭВМ будет сообщать оператору необходимые данные по запросу. Рекомендуемые параметры режима и обнаруженные предаварийные ситуации высвечиваются на сигнальных табло и используются как совет оператору.

Реализация рассчитанных на микроЭВМ управляющих воздействий на начальной стадии разработки комплекса сохраняется за человеком-оператором, который может отвергнуть «совет» микроЭВМ.

Автоматизированный вычислительный комплекс (см. рисунок) состоит из системы контроля и управления процессом бурения СКУБ, выпускаемой Ивано-Франковским ПО «Геофизприбор» (или системы АБУБ [4]), системы цифрового представления информации (СЦПИ) о процессе бурения и микроконтроллера на базе микроЭВМ «Электроника 60М». Длительность цикла опроса датчиков, заданная программно, составляет 0,1 с — 10 мин, в связи с этим требуемое время отклика должно быть не более 50 мс. Точность вычислительных операций составляет 10^{-5} .

СЦПИ состоит из p идентичных независимых модулей, где p — количество контролируемых параметров.



Структурная схема автоматического измерительно-вычислительного комплекса для супервизорного управления процессом бурения нефтяных и газовых скважин с использованием микроЭВМ:

РПД — регулятор подачи долота; РЧВ — регулятор частоты вращения долота; РПА — регулятор расхода бурового раствора; F, W, Q — управляющие воздействия: нагрузка на долото, частота вращения, расход бурового раствора соответственно; СКУБ — система контроля параметров бурения; АЦП1—АЦПп — аналого-цифровые преобразователи; БМ1—БМп — блоки масштабирования; БСЗС1—БСЗСп — блоки световой и звуковой сигнализации; ТКП — табло контролируемых параметров процесса бурения; ЛУ — логическое устройство; ТАС — табло аварийных ситуаций; Р — разветвитель; УВДС — устройство ввода дискретных сигналов; УВАСУ — устройство вывода аналоговых сигналов управления; ТРП — табло рекомендуемых параметров бурения; БИ1 — БИп — блоки индикации

Благодаря модульному построению система сохраняет работоспособность при выходе из строя одного или нескольких блоков.

Аналого-цифровой преобразователь АЦП выполнен на основе цифро-аналогового преобразователя К572ПА1А. Входной сигнал 0...10 В преобразуется АЦП в двенадцатиразрядный двоичный код, который запоминается буферной памятью. С выхода буферной памяти этот код подается на арифметическое устройство. Чтобы обеспечить высвечивание реального значения параметра, необходимо умножить цифровое значение входного напряжения на определенный для данного параметра масштабный коэффициент, который устанавливается при помощи кнопок в счетчике коэффициента оператором до начала работы. Коэффициент масштабирования может изменяться от 1 до 999.

Результат перемножения с выхода арифметического устройства поступает в результирующий счетчик. Эта информация запоминается буферной памятью и выдается в блок информации.

Звуковая и световая сигнализации контролируют выходы параметра за максимальное и минимальное значения.

После записи текущего значения параметра (в счетчик текущего значения) в счетчики минимального и максимального значений коэффициента заносятся соответствующие значения параметра. Логическое устройство ЛУ определяет отношение объемов этих счетчиков. Выходной сигнал ЛУ включает звуковую и световую сигнализацию.

Блок управления содержит кварцевый генератор и цифровой автомат. Он выполняет роль синхронизатора и управляет работой всего модуля.

Питание модулей осуществляется стабилизированным напряжением 5 В.

СЦПИ выполнена на микросхемах серии К155.

Техническая характеристика СЦПИ

Пределы изменения параметров процесса бурения, преобразованные в аналоговый сигнал, В	0...10
Основная приведенная погрешность преобразования, не более, %	±1,0
Рабочая область температур, °С	-10...+70
Индикация результатов измерения	Цифровая
Напряжение питания, В	220
Частота, Гц	50±5
Мощность, потребляемая одним каналом, ВА	15
Масса одного модуля, кг	3
Габаритные размеры модуля, мм	280×235×70

СЦПИ в сочетании с системой контроля и управления СКУБ — основа для создания объектового микроконтроллера.

МикроЭВМ сопрягается с СЦПИ посредством разветвителя Р. Кроме объектовых периферийных устройств

к микроЭВМ подключаются устройства ввода дискретных сигналов УВДС и вывода аналоговых сигналов управления УВАСУ и табло рекомендуемых параметров режима бурения (ТРП). Унифицированные сигналы 0...10 В от СКУБ через интерфейс ввода-вывода поступают в микропроцессор. Одна часть сигналов используется постоянно для контроля и отображения на ТРП, другая — сигналы, поступающие от периферийных устройств или оператора, и реального времени от таймера — для расчетов. Результаты расчетов оптимальных параметров бурения и времени бурения выводятся на табло рекомендуемых параметров ТРП и через устройство вывода аналоговых сигналов управления УВАСУ подаются на регуляторы подачи долота РПД, частоты вращения ротора РЧВ и расхода бурового раствора РР.

Созданный комплект датчиков применен в экспериментальных образцах микропроцессорных систем АБУБ-2, АБУБ-3, «Карпаты», эксплуатирующихся в течение нескольких лет на буровых предприятиях Прикарпатья. Устройство контроля момента на валу буровых механизмов рекомендовано к серийному производству. Алгоритмы и программы оптимизации процесса бурения внедрены в АСУ ТП бурения Крестиненского УБР ПО «Укрбургаз» и позволяют управлять выбором оптимальных нагрузки на долото и частоты вращения ротора, а также времени пребывания долота на забое скважины. Благодаря системе управления стоимость метра проходки скважины снижена на 6—10 руб.

ЛИТЕРАТУРА

1. Терехов Н. И. Автоматическое регулирование и управление режимами бурения. — М.: Недра, 1982. — 205 с.
2. Демихов В. И., Леонов А. И. Контрольно-измерительные приборы при бурении скважин. — М.: Недра, 1980. — 304 с.
3. Система геолого-технологического контроля процесса бурения «Карпаты» / Г. Н. Семенцов, Н. М. Саух, Я. Р. Когуч и др. — Инф. листок. — Ивано-Франковск: ЦНТИ, 1984. — 4 с.
4. Аналоговое специализированное вычислительное устройство контроля параметров процесса бурения АБУБ-3 / Г. Н. Семенцов, Н. М. Саух, Я. Р. Когуч. — Ивано-Франковск: ЦНТИ, 1984. — 4 с.
5. Козловский Е. А., Питерский В. М., Комаров М. А. Кибернетика в бурении. — М.: Недра, 1982. — 298 с.
6. Семенцов Г. Н. Математическое описание процесса углубления скважин. — Разведка и разработка нефтяных и газовых месторождений. — Львов: Вища школа, 1981, вып. 18, РМНТС, с. 64—68.

Статья поступила 5 июня 1985 г.

УДК 681.3.072

ОБМЕН ОПЫТОМ

Ю. Д. Бужан, И. Н. Осипов

ПАНЕЛЬ УПРАВЛЕНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМОЙ

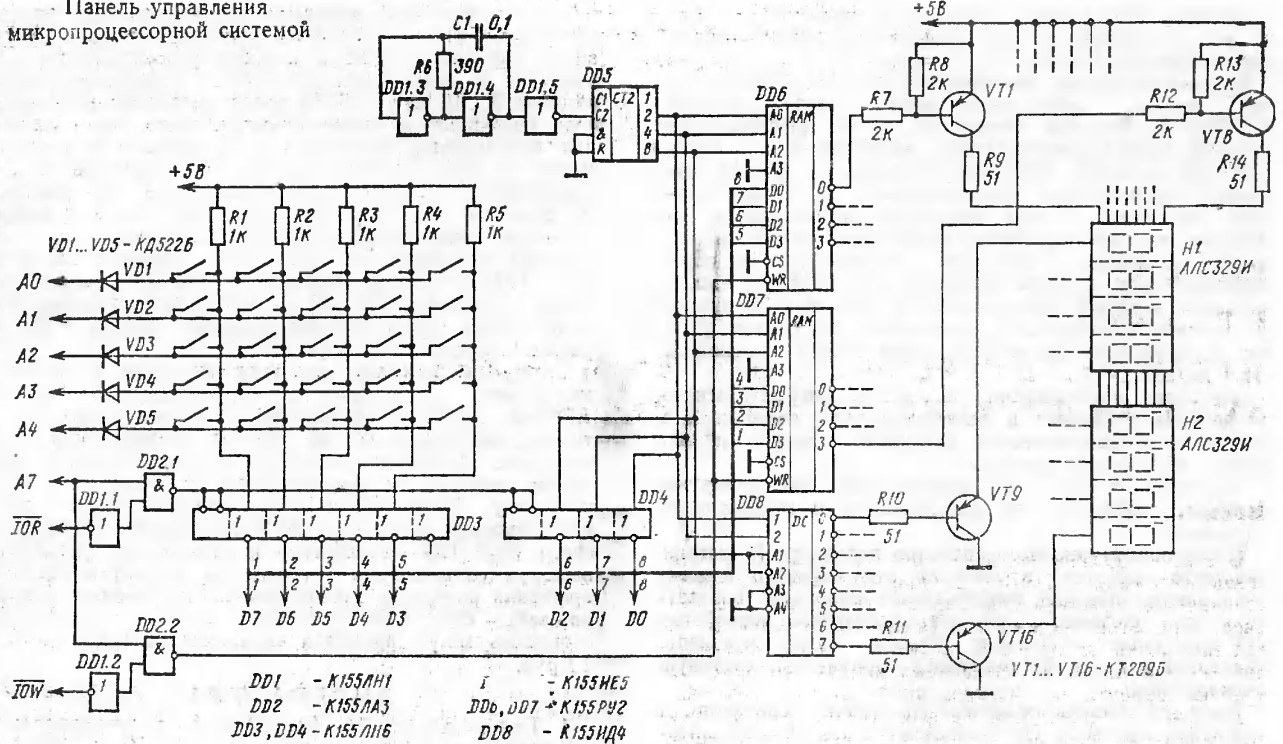
При применении стандартных дисплеев в некоторых случаях удобно использовать специализированную панель управления, имеющую 8-разрядный знаковый светодиодный индикатор и клавиатуру, позволяющую вводить шестнадцатеричные цифры и задавать необходимые режимы работы устройства.

Панель управления (см. рисунок), примененная в контрольно-измерительном прибо-

ре с встроенным микропроцессором (МП) КР580ИК80А, содержит матричную клавиатуру и устройство динамической индикации на семисегментных светодиодных индикаторах.

Клавиатура представляет собой матрицу ключей 5×5. Горизонтальные линии подключены к шине адреса (разряды А0...А4), вертикальные — через порт ввода (микросхемы DD3, DD4) к шине данных (разряды D3...D7).

Панель управления
микропроцессорной системой



Порт выбирается при появлении на шине адреса комбинации 1XXXXXXX (X — безразличное состояние). При линейной выборке устройств ввода-вывода объем аппаратной части уменьшается и сужается адресное пространство, что во многих случаях несущественно.

Клавиатура работает следующим образом: МП выполняет команду ввода из порта с адресом 1XX11110 и анализирует состояние разрядов D3...D7. Если при этом нажат один из ключей, связанных с линией A0, то в соответствующем разряде будет «Лог. 1» и МП перейдет к подпрограмме, соответствующей функциональному назначению ключа, в противном случае МП переходит к выполнению команды ввода из порта с адресом 1XX11101, затем с адресом 1XX11011 и т. д. Таким образом, осуществляется сканирование клавиатуры.

Устройство динамической индикации содержит генератор (микросхема DD1.3...DD1.5), счетчик (микросхема DD5), ОЗУ 16 слов × 8 разрядов (микросхемы DD6, DD7), дешифратор (микросхема DD8), транзисторные ключи и светодиодные индикаторы. Состояние счетчика, определяющее адрес ячейки ОЗУ, выдается на шину данных через порт ввода (разряды D0...D2). Так как ОЗУ и дешифратор работают от одного генератора, есть однозначное соответствие между ячейками памяти и индикаторами, т. е. каждой из восьми младших ячеек ОЗУ соответствует один из восьми

индикаторов, поэтому последние воспроизводят информацию, записанную в ОЗУ.

Информация на индикаторы выводится, когда МП выполняет команду ввода из порта с адресом 1XXXXXXX и анализирует состояние разрядов D0...D2 (частота регенерации индикатора значительно меньше максимальной частоты обращения процессора). Этот процесс повторяется до появления необходимого адреса ячейки ОЗУ (номера индикатора), после чего МП выполняет команду вывода информации в порт с адресом 1XXXXXXX. По этой команде в ячейку ОЗУ записывается информация, отображаемая соответствующим этой ячейке индикатором.

Статья поступила 25 марта 1985 г.

РЖ ВИНТИ, АВТ-85

8.81.354. Управление роботом с использованием микроЭВМ. 1984 (англ.; рез. нем.)

Рассмотрены требования и ограничения, которые должны быть учтены при разработке систем для программирования промышленных роботов (ПР) и управления (У) ими. Описаны подходы к описанию кинематики и динамики ПР, методы программирования ПР и алгоритмы У движениями ПР. Предложена структура гибкой распределенной системы У на основе сети микроЭВМ. Иерархическая система У построена по модульному принципу и включает микроЭВМ, осуществляющие У на различных уровнях (У отдельными степенями подвижности, их координации, обработки и учета при У сенсорной информации, формирования движений ПР). Предусмотрено сопряжение системы с вычислительными средствами САПР и АСУ ТП.

Переходить на комплектную поставку технологических систем и комплексов машин. Всемерно развивать фирменный ремонт и обслуживание силами изготовителей сложной и особо точной техники...

Из проекта Основных направлений экономического и социального развития СССР на 1986—1990 годы и на период до 2000 года

УДК 681.322.1

Г. Р. Громов

НАДЕЖНОСТЬ ПЕРСОНАЛЬНЫХ ЭВМ И ПРОИЗВОДИТЕЛЬНАЯ ЗАГРУЗКА ФИРМЕННОЙ СЕТИ СЕРВИСА

Введение

В статье сделана попытка оценить количественную связь надежности ПЭВМ с относительной нагрузкой, создаваемой действующим парком машин на сеть сервиса. При этом надежность ПЭВМ измеряется в часах наработки на отказ, а загрузка сети сервиса — долей общего числа произведенных машин, которые требуют ремонта за фиксированный отрезок времени.

На рис. 1 приводится график интенсивности отказов массовых изделий радиоэлектроники. Традиционно на нем выделяют три основных этапа эксплуатации. Первый — с высокой интенсивностью отказов — получил название — период «выгорания». В этот период «выжигаются» дефектные микросхемы, проявляются наиболее грубые погрешности изготовления и сборки изделий электромеханики и т. д. Только после устранения всех этих неисправностей наступает второй этап — период активной эксплуатации изделия. В общем случае предполагается, что за это время интенсивность отказов меняется слабо*. Третий этап — рост интенсивности отказов из-за физического износа оборудования.

В дальнейшем изложении предполагается, что:

1) на предприятиях, производящих ПЭВМ, приняты необходимые меры, чтобы период «выгорания» был в основном завершен еще на цеховых испытательных

стендах, где отдельные узлы и блоки, а затем и машину в целом «гоняют» для этого в условиях предельных напряжений питания; на «тепло» и «влагу»; вибрацию и т. д.;

2) моральный износ ПЭВМ наступает значительно раньше их физического износа.

В случае, когда указанные выше предположения имеют под собой реальные основания, можно ожидать, что за период эксплуатации ПЭВМ интенсивность отказов окажется близкой к постоянной и, соответственно, вероятность отказа будет характеризоваться экспоненциальным законом распределения. До сих пор экспоненциальное распределение применялось, в основном, в расчетах надежности той части военной и аэрокосмической техники, где предположения (1) и (2), как правило, выполняются [1]. Как уже неоднократно отмечалось, надежды на экономическую эффективность массового внедрения ПЭВМ в народное хозяйство теряют под собой всякую почву, если не будут приняты необходимые организационно-технические меры для обеспечения промышленностью, по крайней мере, столь же высокого уровня надежности* и для ПЭВМ [2, 3].

Оценка экономической дилеммы распределения ресурсов: «надежность ПЭВМ — сеть сервиса»

Известно, что в настоящее время большая часть неисправностей действующего парка ЭВМ устраняется непосредственно бригадами обслуживающего их персонала. Контакты со специализированной службой сервиса ограничиваются, как правило, лишь случаями достаточно тяжелых неисправностей или необходимостью получения ЗИПа. В отличие от этого сложившегося у нас в стране за последние десятилетия парка «самообслуживаемых» ЭВМ, машины индивидуального пользования — ПЭВМ — идут с завода-изготовителя непосредственно на рабочие столы специалистов, ранее незнакомых с вычислительной техникой. Поэтому ситуация здесь складывается принципиально иная — практически любая неисправность — повод к рабочему контакту со службой сервиса.

* Как отмечалось в работе [2], согласно одному из последних определений, «ПЭВМ — это вычислительная машина с надежностью военной аппаратуры и ценой изделия бытовой электроники».

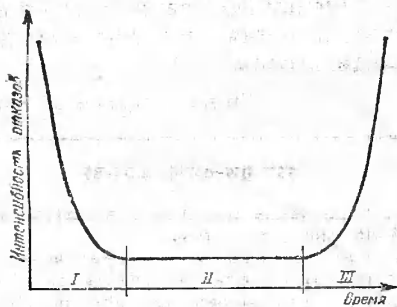


Рис. 1. Три периода эксплуатации:

I — «выгорание»; II — постоянная интенсивность отказов; III — «старение»

* Практически это зависит, разумеется, от многих, в том числе и не всегда учитываемых традиционными моделями надежности [1] факторов, которые определяются, в основном, конкретными областями приложений и соответственно различными условиями эксплуатации ЭВМ.

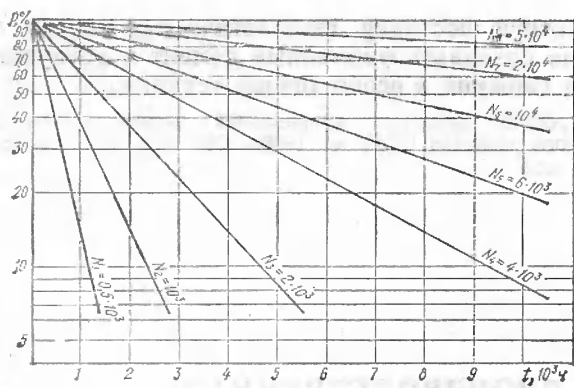


Рис. 2. Относительная доля P [%] парка машин, сохраняющих работоспособность за время эксплуатации t [ч], для различных значений показателя надежности N (наработка на отказ)

Примечание. Рис. 2 выполнен для экспоненциального закона распределения, в упрощающем качестве анализ предположении, что $P(t) = e^{-t/N}$ (см., например, [1, с. 15])

Таким образом, вероятность отказа ПЭВМ и вероятность поступления заявки в службу сервиса для парка ПЭВМ следует считать численно равными.

Из рис. 2, выполненного для экспоненциального закона распределения вероятности отказов, можно оценить, какая часть ПЭВМ останется работоспособной — не испытает ни одного отказа (т. е. не потребует внимания со стороны службы сервиса) за тот или иной отрезок времени при различных значениях надежности выпускаемых ПЭВМ.

Известно, что в условиях роста тиража массово выпускаемых изделий микроэлектроники общая стоимость начальных затрат на исследование и разработку того или иного конкретного технологического решения (в данном случае решения, повышающего надежность ПЭВМ), приведенная к стоимости единичного изделия, быстро снижается.

В то же время «цена устранения одного отказа» изделия за период эксплуатации снижается (даже там, где она действительно снижается...) весьма медленно, так как уровень автоматизации работ, выполняемых в мастерских сети сервиса, неизбежно значительно отстает от уровня автоматизации заводского производства. Кроме того, непреодолимым барьером для снижения средней «цены единичного отказа» ПЭВМ остается стоимость доставки машины в пункт обслуживания (или наоборот, вызова представителя службы сервиса). Поэтому с расширением масштабов массового производства ПЭВМ экономически целесообразно соответственно наращивать долю материально-технических ресурсов, направляемых на решение первоочередной, наиболее острой из всех задач массовой компьютеризации — повышение эксплуатационной надежности выпускаемых машин.

Техника оценки

Связь годовой производственной загрузки сети сервиса с надежностью парка выпускаемых ПЭВМ показана на рис. 3. Следует отметить, что переход от оси времени на рис. 3 к «астрономическому» времени, т. е. от суммарного времени эксплуатации нерегулярно включаемого в работу индивидуального инструмента, каким является ПЭВМ, к «астрономическому» времени (например, для календарного планирования загрузки сети сервиса) — непростая операция.

В зависимости от типа и конфигурации ПЭВМ перемены в работе различным образом влияют на интен-

сивность отказов. Например, как показывает опыт работы действующих в настоящее время вычислительных центров, интенсивность отказов твердотельных блоков электронной аппаратуры весьма заметно повышается после любого перерыва в работе машины и, кроме того, существенно связана с длительностью этого перерыва. Наиболее благоприятны для блоков электроники условия длительной круглосуточной работы, когда практически отсутствуют неизбежные при любых запусках машины резкие колебания напряжения питания; остаются относительно постоянными температурный режим, влажность и другие параметры эксплуатации. Поэтому, в первом приближении, можно считать, что для блоков твердотельной электроники резкое «ускорение бега часов» по шкале старения аппаратуры после перерыва в работе машины компенсирует (а нередко и значительно перекрывает) «остановку часов старения» на период выключения машины, т. е. эквивалентное время старения составит 9 тыс. ч за календарный год.

В то же время электромеханические внешние устройства ЭВМ (диски, печатающие устройства и др.) подвергаются физическому износу, естественно, лишь в движении. Поэтому в «холодном» состоянии «часы времени эксплуатации» для электромеханических устройств почти полностью останавливаются* (если не учитывать рост интенсивности отказов от пребывания в «холодном» состоянии электронных блоков электромеханических устройств — контроллеров).

Оценки границы разброса значений на графиках, отражающих связь вероятности безотказной работы с показателем надежности ПЭВМ. Пусть рабочая загрузка парка ПЭВМ класса, например, ДВК-2 характеризуется средним временем полезной работы 3 ч за полный рабочий день, или около 900 ч суммарного времени эксплуатации за календарный год (все остальное время машина, как предполагается, выключена). Для ЭЛТ дисплея и электромеханических устройств ПЭВМ такой режим эксплуатации означает ≈ 900 «часов старения» за год работы машины у пользователя. Таким образом, в рамках принятых выше предположений об ожидаемом характере эксплуатации ПЭВМ, «темп старения» твердотельных блоков, оцениваемый на уровне 9 тыс. ч старения в год, окажется в 10 раз выше, чем соответствующий темп для ЭЛТ и электромеханических устройств. Для других электромеханических устройств: принтеров, графопостроителей, дигитайзеров и т. д. разница в темпе старения окажется еще больше, так как в среднем они включаются лишь на небольшую часть от общего времени работы ПЭВМ.

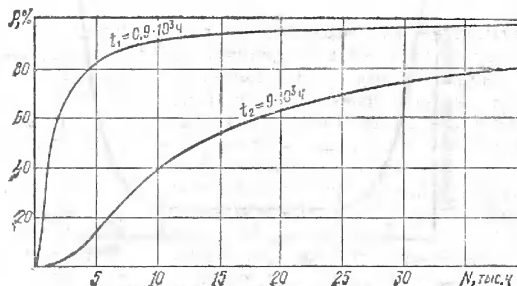


Рис. 3. Увеличение доли P [%] парка работоспособных машин с ростом показателя надежности N для двух значений времени эксплуатации: $t_1 = 900$ ч, $t_2 = 9$ тыс. ч

* Как, по-видимому, в значительной степени приостанавливается ход «часов старения» на время пока выключены электровакуумные приборы, например электроно-лучевая трубка (ЭЛТ) дисплея.

Итак, в первом приближении можно считать, что оценку вероятности безотказной работы твердотельных блоков ПЭВМ можно выполнять по нижней кривой рис. 3, построенной в предположении, что «темпы старения» ПЭВМ составляет 9 тыс. ч в год, тогда как для ЭЛТ и электромеханических блоков — по верхней, соответствующей «темпу старения», — 900 ч в год.

Пример

Пусть необходимо выполнить приближенную оценку ожидаемой нагрузки на сеть сервиса для парка профессиональных ПЭВМ, выпускаемых в следующей конфигурации:

- микроЭВМ (процессор, ОЗУ, ПЗУ, контроллеры и бесконтактная клавиатура) с надежностью $N_1=20$ тыс. ч;
- ЭЛТ с надежностью $N_2=15$ тыс. ч, наработки на отказ;
- гибкий диск — $N_3=10$ тыс. ч;
- принтер — $N_4=5$ тыс. ч.

По нижней кривой рис. 3 находим, что для показателя надежности $N_1=20$ тыс. ч вероятность безотказной работы в течение года твердотельных блоков для микроЭВМ составляет $p_1=0,68$.

По верхней кривой находим вероятности безотказной работы остальных блоков: для ЭЛТ $p_2=0,95$; для диска $p_3=0,93$; для принтера $p_4=0,84$.

В предположении, что вероятности выхода из строя отдельных блоков статистически независимы, находим вероятность безотказной в течение года работы такой ПЭВМ: $p=p_1 \cdot p_2 \cdot p_3 \cdot p_4 \approx 0,5$.

Таким образом, приближенная оценка вероятности безотказной работы ПЭВМ, вышолненная по рис. 3 с учетом показателей надежности отдельных блоков, и приведенных выше предположений о характере их эксплуатации, дает основания ожидать, что около половины из общего парка таких ПЭВМ в течение года эксплуатации потребуют внимания фирменной службы сервиса.

Примечание. Используемые в данном примере численные значения характеристик надежности отдельных блоков ПЭВМ не имеют отношения к реальным параметрам какой-либо из выпускаемых в настоящее время машин, а приводятся здесь с единственной целью проиллюстрировать возможную технику оценки при работе с графиками типа рис. 3.

Итак, системный коэффициент типа старения (КТС), связывающий календарное время и время активной эксплуатации машины, будет заметно отличаться для разного типа ПЭВМ и различных конфигураций одного типа ПЭВМ.

Разработка корректных методов оценки численного значения КТС для массовых изделий микропроцессорной техники, в том числе для основных классов выпускаемых в настоящее время ПЭВМ, — одна из актуальных задач, которые ставит перед наукой о надежности практика массового внедрения микроЭВМ. Решение этой задачи позволило бы, в частности, указывать в техническом паспорте на ПЭВМ отдельно две практически полезные характеристики надежности:

- 1) для времени активной эксплуатации изделия;
- 2) для периода длительных пауз в его работе. В настоящее время именно так указывается гарантийный срок для значительной части радиоэлектронной аппаратуры — раздельно: на период хранения изделия и на время его активной эксплуатации (по существу, те же два типа «старения»).

В общем случае, до получения уточненных значений КТС для основных классов машин используемого парка ПЭВМ, рабочие оценки загрузки фирменной сети сервиса, выполняемые по рис. 3, носят ориентировочный характер и должны корректироваться с накоплением опытных данных.

Заключение

Текущая крутизна характеристики «надежность — сервис» позволяет оценить, насколько заметно отражается на производственной загрузке сети сервиса изменение надежности машины. Например, как видно из рис. 3, после достижения такого уровня надежности ПЭВМ, при котором 70—80% машин уже не требует внимания службы сервиса в течение года их эксплуатации, дальнейшее повышение надежности все слабее отражается на изменении загрузки сети сервиса. Не останавливаясь здесь на деталях разработки формального аппарата поиска экономически оптимальной стратегии распределения ресурсов между задачами развития сети сервиса и повышения надежности выпускаемых машин, отметим лишь, что с ростом тиража выпуска машин точка «экономического оптимума» на характеристике «надежность — сервис» перемещается вправо-вверх, так как с увеличением парка машин соответственно растет экономическая цена «процента отказов». И наоборот — для малых серий ПЭВМ точка экономического оптимума оказывается в области меньших значений показателя надежности на интервале более высокой крутизны характеристики «надежность — сервис».

ЛИТЕРАТУРА

1. Лонгботтом Р. Надежность вычислительных систем. — М.: Энергоатомиздат, 1985. — 283 с.
2. Громов Г. Р. Профессиональные приложения персональных ЭВМ. — Микропроцессорные средства и системы, 1985, № 3, с. 9—15.
3. Громов Г. Р. Макс-проблема микропроцессора. — Известия, 20 июля 1985 г. (№ 201), с. 2.

Статья поступила 16 октября 1985 г.

КОНФЕРЕНЦИЯ В МГУ

Всесоюзная конференция «Психологические проблемы создания и использования ЭВМ» проходила в Москве на базе факультета психологии МГУ 25—27 сентября 1985 года. В конференции приняли участие психологи, философы, социологи, экономисты, математики, специалисты по информатике и вычислительной технике.

На открытии конференции выступили проректор МГУ профессор В. А. Садовничий, академик А. Н. Тихонов и академик АПН СССР А. А. Бодалева. Они отметили место вычислительной техники в решении поставленных партией задач ускорения научно-технического прогресса, актуальность постановки психологических проблем развития информатики.

Член-корреспондент АПН СССР В. М. Монахов дал характеристику

психолого-педагогических проблем компьютеризации образования. Академик АПН СССР Е. Н. Соколов раскрыл содержание нейробионического подхода к созданию искусственного интеллекта. Член-корреспондент АПН СССР Н. Ф. Талызина обосновала положение о том, что наибольшая эффективность компьютеризации достигается при опоре на деятельностную теорию обучения. Автор этой заметки дал характеристику прикладных, теоретических и методических проблем психологической науки, возникающих в условиях ее взаимодействия с информатикой.

Значительный интерес вызвали лекции, с которыми выступили член-корреспондент АН СССР Б. Ф. Ломов «НТР и средства умственного развития человека», член-корреспондент АН УССР А. Г. Иващенко «Индуктив-

ный подход к созданию искусственного интеллекта и задачи психодиагностики». Программа конференции включала в себя работу четырех секций: «Психология, информатика, искусственный интеллект»; «Методы и принципы психологического исследования и обеспечения деятельности программистов и пользователей ЭВМ»; «Психологические и психолингвистические принципы организации диалога между человеком и ЭВМ»; «Психологические проблемы практического использования систем различного назначения».

В период работы конференции была организована дискуссия на тему: «Модели психики в системах искусственного интеллекта; творчество и алгоритмы».

Материалы конференции изданы в виде сборника объемом 15 п. л. (издательство МГУ).

О. К. Тихомиров

УДК 681.3.02

Малашевич Б. М., Романов Ф. И. Перспективная конструкционная система микропроцессорных средств.— Микропроцессорные средства и системы, 1985, № 4, с. 3.

Рассмотрена проблема унификации конструкций МСС, проведен анализ областей эффективности унификации с технической и экономической точек зрения. На основе анализа наиболее распространенных отечественных и зарубежных конструктивных систем сформулированы требования к перспективной унифицированной конструкционной системе МСС и сделан ее выбор.

УДК 681.3

Кузнецов С. О., Прохоров Н. Л., Раев В. К. «Электронный диск» на цилиндрических магнитных доменах.— Микропроцессорные средства и системы, 1985, № 4, с. 11.

Анализируется опыт эмуляции НГМД с помощью ЦМД-микросхем программным, аппаратным и программно-аппаратным способами.

УДК 681.3.06

Лавров С. С., Слисенко А. О., Цейтин Г. С. Проект плана-программы по специальности «Информатика и системное программирование».— Микропроцессорные средства и системы, 1985, № 4, с. 20.

Обосновывается необходимость введения в вузах специальности «Информатика и системное программирование», предлагается проект учебного плана и программы дисциплин.

УДК 681.3:657.47

Бетелин В. Б. Проблема автоматизации подготовки программной документации.— Микропроцессорные средства и системы, 1985, № 4, с. 29.

Предлагается модель программной документации, описывается специализированный редактор текстов, реализованный на ее основе.

УДК 681.3.066

Щелкунов Н. Н., Дианов А. П. Программирование микросистем реального времени.— Микропроцессорные средства и системы, 1985, № 4, с. 31.

Рассмотрен многозадачный режим работы микросистем в реальном масштабе времени. Предложен состав и реализация простого ядра системы реального времени.

УДК 681.3.06, 681.2

Лукьянов Д. А. Как написать кросс-транслятор с языка ассемблера.— Микропроцессорные средства и системы, 1985, № 4, с. 35.

Рассмотрен способ построения кросс-трансляторов с языка ассемблера для микропроцессоров с разрядностью адреса до 16, предназначенных для эксплуатации на ЭВМ класса СМ-4 и «Электроника 60». Кросс-ассемблер оформляется как список макроопределений, а для трансляции прикладной программы используется уже существующий макроассемблер МАКРО-11. Приведен полный текст кросс-ассемблера МИКРОСС для микропроцессора КР580ИК80.

UDC 681.3.02

Malashevich B. M., Romanov F. I. Perspective Microprocessor Device System.— Microprocessor Devices and Systems, 1985, N 4, p. 3.

The problem of unification is considered for mechanical standard of microprocessor devices. The technical and economical analysis is presented. The requirements for a unified system are formulated.

UDC 681

Kuznetsov S. O., Prochorov N. L., Raev V. K. Floppy Disk Drive Emulation on a Cylinder Magnetic Domain Memory.— Microprocessor Devices and Systems, 1985, N 4, p. 11.

The paper analyzes the experience of software and hardware emulation of FDD with CMD devices.

UDC 681.3.06

Lavrov S. S., Slisenko A. O., Tseitlin G. S. The Project of Programme on "Informatics and Systems Programming".— Microprocessor Devices and Systems, 1985, N 4, p. 20.

The authors discuss the necessity of studying the speciality "Informatics and Systems Programming" in the high schools. The lecture plan and programme project is proposed.

UDC 681.3:657.47

Betelin V. B. On the Problem of Automatic Program Documentation Development.— Microprocessor Devices and Systems, 1985, N 4, p. 29.

The program documentation model is proposed and a special text editor is discussed.

UDC 681.3.066

Schelkunov N. N., Dianov A. P. Programming Real Time Systems.— Microprocessor Devices and Systems, 1985, N 4, p. 31.

The paper considers multitasking operation of microsystems in a real time. The structure and implementation of a simple kernel of such a system is considered.

UDC 681.3.06.681.2

Lukyanov D. A. A Write-it-Yourself Cross-Assembler.— Microprocessor Devices and Systems, 1985, N 4, p. 35.

A simple method of cross-assembler building for microprocessors with up to 16-bit addressing is discussed. Based on using already existing MACRO-11 macroassembler tailored for microprocessor instruction set with unique macrodefinition block, it is intended for use on PDP-11 compatible mini- and microcomputers. The same method is suitable with M80 assembler under CP/M. The complete macrodefinition listing for KR580IK80 is given as illustration.

УДК 681.3.06:322.1

Липаев В. В., Каганов Ф. А., Керданов А. В., Загубин Ю. В., Колдобский А. Э., Новичков А. С. Система автоматизации проектирования программ на базе персональных ЭВМ (система ПРА).— Микропроцессорные средства и системы, 1985, № 4, с. 42.

Описаны особенности ПРОМЕТЕЙ-технологии, обеспечивающей автоматизацию разработки программ встраиваемых мини- и микроЭВМ. Показаны пути использования персональных ЭВМ (ПЭВМ) по поддержке ПРОМЕТЕЙ-технологии. Представлены функции и конструкция настраиваемой кросс-системы ПРА, ориентированной на разработку программ на ПЭВМ. Система ПРА может использоваться для создания программ различных микроЭВМ, в том числе построенных на основе микропроцессора K1810BM86.

УДК 681.3.06

Вельбицкий И. В., Ковалев А. Л. Графический стиль программирования для персональной ЭВМ.— Микропроцессорные средства и системы, 1985, № 4, с. 46.

Рассматривается отечественная R-технология программирования и средства ее инструментальной поддержки на персональной ЭВМ «Электроника НЦ-80-20» и микроЭВМ «Электроника 60». Отличительными чертами R-технологии являются графический способ представления алгоритмов и программ, проектирование программ в неформальных обозначениях, автоматизация управления разработкой программ.

УДК 621.3.049.17.022.72

Адамов Е. О., Дукарский С. М., Корягин Д. А. Интегрированная машиностроительная система КАПРИ.— Микропроцессорные средства и системы, 1985, № 4, с. 53.

Описаны принципы построения, структура и функции интегрированной машиностроительной системы КАПРИ, создаваемой и частично реализованной с целью существенного сокращения сроков создания экспериментальных установок и стендов, а также решения социальных задач сокращения трудопотребления и ликвидации потребности в работниках социально-непривлекательных профессий. Приведены результаты исследований и разработок компонент системы КАПРИ.

УДК 007.52:681.3

Адамов Е. О., Гримайло С. И., Камынин С. С., Корягин Д. А., Смольянов Ю. П. Роботизированный механообрабатывающий комплекс для участка гибкого производства.— Микропроцессорные средства и системы, 1985, № 4, с. 57.

Описываются аппаратно-программные средства роботизированного комплекса, управляемого от микроЭВМ. Основу комплекса составляет токарный станок 16K20T1 с системой управления «Электроника НЦ-31» и робот РК-1. Комплекс разработан в институте атомной энергии им. И. В. Курчатова совместно с институтом прикладной математики им. М. В. Келдыша.

УДК 681.32-181.4

Савин А. И., Кузнецов М. Н., Уксусов А. С., Целяпин А. Н. Структура системы управления ГАП.— Микропроцессорные средства и системы, 1985, № 4, с. 62.

Даны назначение и структура системы управления ГАП на базе станков с ЧПУ, промышленной робототехники, автоматических складов, транспортных систем и интеллектуальных подсистем управления на основе микро- и мини-ЭВМ.

UDC 681.3.06:322.1

Lipaev V. V., Kaganov F. A., Kerdanov A. V., Zagubin Yu. V., Koldobsky A. E., Novichkov A. S. Program Development Automation System on the Basis of Personal Computers (System PRA).— Microprocessor Devices and Systems, 1985, N 4, p. 42.

The paper describes peculiarities of the PROMETEY-technology which provides program development automation for embedded mini- and microcomputers. The possibilities of using professional personal computers for supporting PROMETEY-technology are considered. The functions and architecture of the adaptable cross-system are presented. The PRA system is intended for creation of various microcomputer programs including the K1810BM86-based microcomputers.

UDC 681.3.06

Velbitsky I. V., Kovaljov A. L. Graphical Style of Programming for Personal Computer.— Microprocessor Devices and Systems, 1985, N 4, p. 46.

The paper considers R-technology of programming with application to the microcomputers "Electronica NC 80-20" and "Electronica 60". R-technology is based on the graphical representation of algorithms and programs providing the convenient means of program development.

UDC 621.3.049..17.022.72

Adamov E. O., Dukarsky S. M., Korjagin D. A. Intergrated Machine-Building System KAPRI.— Microprocessor Devices and Systems, 1985, N 4, p. 53.

The paper describes the principles, functions and architecture of intergrated system KAPRI, intended for automated development of machine-building tools and equipment. The system will speed up the development process, reduce the required manpower and eliminate the necessity for socially non-attractive jobs. The system components are discussed.

UDC 007.52:681.3

Adamov E. O., Grimajlo S. I., Kamynin S. S., Korjagin D. A., Smoljanov Yu. P. Robot-Based Metal-Cutting Equipment for a Flexible Manufacturing Division.— Microprocessor Devices and Systems, 1985, N 4, p. 57.

The paper discusses hardware and software components of a special robot-based complex controlled by a microcomputer. The basis of the complex is the lathe 16K20T1 controlled by "Electronica NC-31" and robot RK-1. The complex is developed jointly by Kurchatov Institute of Nuclear Energy and Keldysh Institute of Applied Mathematics.

UDC 681.32-181.4

Savin A. I., Kuznetsov M. N., Uksusov A. S., Tseljapin A. N. FMS Control Structure.— Microprocessor Devices and Systems, 1985, N 4, p. 62.

The paper describes the purpose and the structure of the flexible manufacturing control system. The FMS consists of numerically controlled instruments, industrial robots, automatic storehouses, transport subsystems and intelligent control system implemented on micro- and minicomputers.

УДК 681.3.06-181.4

Соломенцев Ю. М., Сосонкин В. Л. **Оперативное управление гибкими производственными системами.**—Микропроцессорные средства и системы, 1985, № 4, с. 65.

Рассмотрена система оперативного управления гибкой производственной системой, т. е. та часть математического обеспечения локальной микропроцессорной сети, которая относится к управлению в реальном времени. Работа системы оперативного управления ГПС описана в терминах сетевой операционной системы реального времени. Поставлена задача получения универсального решения для различных гибких систем.

УДК 681.3:007.52

Каляев А. В., Чернухин Ю. В., Брюхомицкий Ю. А., Каляев И. А., Носков В. П. **Построение однородных управляющих структур адаптивных автономных роботов.**—Микропроцессорные средства и системы, 1985, № 4, с. 68.

Рассмотрены принципы построения систем управления адаптивными транспортными, манипуляционными роботами и их группами с использованием однородных управляющих структур, а также действующие макеты таких систем. Дается краткое описание большой интегральной схемы однородной управляющей структуры, позволяющей резко сократить аппаратные затраты, повысить надежность и технологичность управляющих систем.

УДК 681.325.5-181.4

Бамбуров Н. С., Екимов А. В., Ермолин Ю. В., Мамаков П. В., Сютин А. Н. **Программируемый групповой контроллер на базе МПК БИС КР580.**—Микропроцессорные средства и системы, 1985, № 4, с. 78.

Рассмотрен микроконтроллер на базе МПК БИС КР580, обеспечивающий доступ нескольким пользователям к высокопроизводительной ЭВМ по одной терминальной линии связи, а также управление несколькими микроЭВМ от одного терминала.

УДК 681.32:621.316.7

Михайлов С. А. **Система управления электроприводами судовых грузовых комплексов.**—Микропроцессорные средства и системы, 1985, № 4, с. 83.

Показана эффективность микропроцессорного управления электроприводами грузовых комплексов при ограниченной мощности судовой многоагрегатной электростанции. Предложены алгоритмы, программное обеспечение для микроЭВМ и схемные решения локальных автоматов.

УДК 622.24.007:681.322

Семенов Г. Н. **Автоматизированный комплекс на базе микроЭВМ «Электроника 60М» для управления бурением.**—Микропроцессорные средства и системы, 1985, № 4, с. 86.

Рассматривается структура автоматического измерительно-вычислительного комплекса для супервизорного управления процессом бурения нефтяных и газовых скважин с использованием микроЭВМ «Электроника 60М».

УДК 681.322.1

Громов Г. Р. **Надежность персональных ЭВМ и производственная загрузка фирменной сети сервиса.**—Микропроцессорные средства и системы, 1985, № 4, с. 89.

Оценивается количественная связь надежности персональных ЭВМ с нагрузкой, создаваемой парком машин на сеть сервиса.

UDC 681.3.06-181.4

Solomentsev Yu. M., Sosonkin V. L. **Operative Control of Flexible Manufacturing Systems.**—Microprocessor Devices and Systems, 1985, N 4, p. 65.

The paper discusses flexible manufacturing control system. The software for LAN control in a real time is considered. The FMS control is described in terms of real-time network operation system. The problem of developing general approach for various flexible manufacturing systems is discussed.

UDC 681.3:007.52

Kaljaev A. V., Chernukhin Yu. V., Brjuhomitsky Yu. A., Kalyaev I. A., Noskov V. P. **Construction of Homogeneous Control Structures for Autonomous Adaptive Robots.**—Microprocessor Devices and Systems, 1985, N 4, p. 68.

The paper describes the principles of building control systems for adaptive transport and manipulation robots on the basis of homogeneous control structures. LSI implementation of a homogeneous control structure provides considerable reduction of hardware costs, increases reliability and the quality of control systems.

UDC 681.325.5-181.4

Bamburov N. S., Yekimov A. V., Yermolin Yu. V., Mamakov P. V., Sytin A. N. **Programmable Group Controller on the Basis of LSI Set KR580.**—Microprocessor Devices and Systems, 1985, N 4, p. 78.

The authors consider microcontroller which provides the control of several microcomputers from a single terminal. The system is implemented by means of KR580 LSI set.

UDC 681.32:621.316.7

Mikhailov S. A. **Freight Ship Electrical Drive Control System.**—Microprocessor Devices and Systems, 1985, N 4, p. 83.

The author discusses microprocessor control system for electrical drives of freight ships which have limited capacity of a power station. Algorithms, microcomputer software and automata implementation are described.

UDC 622.24.007:681.322

Sementsov G. N. **Automatic Drilling Control System on the basis of "Electronica 60M" Microcomputer.**—Microprocessor Devices and Systems, 1985, N 4, p. 86.

The author considers the structure of automatic measuring and control system for supervisor control of oil and gas well drilling. The system is based on "Electronica 60M" microcomputer.

UDC 681.322.1

Gromov G. R. **PC Reliability and Computer Service Network Efficiency.**—Microprocessor Devices and Systems, 1985, N 4, p. 89.

The author evaluates the quantitative relation between personal computers reliability and the computer service network efficiency.

НОВОСТИ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

Крылов Е. И.—Однокристалльные микроЭВМ серий K1814, K1820, K1816 2

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Белицкий Р. И.—Адресация данных в потоковой мультимикропроцессорной системе с магистральной структурой 1
 Дшуниа В. Л., Борщенко Ю. И., Отрохов Ю. Л., Шишарин С. А.—Одноплатные микроЭВМ ряда «Электроника МС 1201» 2
 Зимовец А. Н., Рыбак К. С., Фомин С. В., Черняк А. Ю., Юрочкин А. Г.—Многомашинная система тестирования микропроцессорных средств вычислительной техники 4
 Златопольский В. Н., Лобов И. Е., Стоянов А. И., Шадрин И. А.—Однокристалльные 4-разрядные микроЭВМ серии K1814 1
 Знаменский Ю. Н., Карев В. В., Крушевский Л. Я., Маслов А. В., Поликарпочкин С. П.—Скоростной дуплексный адаптер межпроцессорной связи «Электроника МС 8002» 1
 Знаменский Ю. Н., Карев В. В., Маслов А. В., Варфоломеев К. Ю.—Средства расширения вычислительных комплексов — «Электроника МС 9506», «Электроника МС 9604», «Электроника МС 4613» 2
 Калошкин Э. П., Румянцев В. И., Сержанович Д. С., Горовой В. В., Харько В. В.—Микропроцессорный комплект БИС серии K583 2
 Кузнецов С. О., Прохоров Н. Л., Раев В. К.—«Электронный диск» на цилиндрических магнитных доменах 4
 Лопатин В. С., Юрочкин А. Г., Баранов Н. Д.—Адаптер магистралей СМ ЭВМ и микроЭВМ «Электроника 60» 1
 Лопатин В. С., Пархоменко П. И., Токмаков В. И.—МикроЭВМ «Электроника МС 1211», «Электроника МС 1212» 2
 Лопатин В. С., Борисенков В. Д., Юрочкин А. Г., Скобленев А. Г., Баранов Н. Д.—Диспетчер памяти микроЭВМ «Электроника МС 1213» 3
 Малашевич Б. М., Романов Ф. И.—Перспективная конструкционная система микропроцессорных средств 4
 Мальцев П. П., Харько В. В.—Особенности структуры центральных процессоров на основе БИС серий K588, K589, K1804 2
 Пархоменко П. И., Бражник В. П.—Унифицированный источник питания для 16-разрядных микроЭВМ 1
 Толстых Б. Л., Еремин С. А.—Однокристалльная 16-разрядная микропроцессорная СБИС повышенной производительности 2

ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

Абрамович С. Н., Бойко В. В., Бутрин Б. П., Казаринов В. Е., Кац М. Я., Кузнецов В. Е.—Профессиональные персональные ЭВМ «Искра 226» 2
 Барышников В. Н., Воронов М. А., Гиглавый А. В., Паначев Ф. И., Романов В. Ю., Титов О. Ф.—Персональная ЭВМ «Ирнша» для кабинетов информатики и вычислительной техники 3

Брендэ В. В., Костиков А. С.—Особенности реализации программных систем на микроЭВМ «Искра 226» 2
 Громов Г. Р.—Профессиональные приложения персональных ЭВМ 3
 Косенков С. М., Полосин А. Н., Сцепицкий З. А., Дябин М. И., Половянюк А. И.—Бытовая персональная микроЭВМ «Электроника БК-0010» 1
 Кочетков Г. Б.—Компьютерные игры: свет и тени 3
 Лысенко Е. Е.—Компьютерная игра с точки зрения психолога 3
 Попов С. Н.—Система диагностики неисправностей микропроцессорной аппаратуры на базе персонального компьютера 1

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Бетелин В. Б.—Проблема автоматизации подготовки программной документации 4
 Вельбицкий И. В., Ковалев А. Л.—Графический стиль программирования для персональной ЭВМ 4
 Громов Г. Р.—Программирование: ремесло, наука, искусство, технология 1
 Ершов А. П.—Алгоритмический язык в школьном курсе основ информатики и вычислительной техники 2
 Карась И. З.—Опыт функционирования промышленного предприятия по производству программных средств 1
 Корсаков С. Я., Крылов В. В., Кочетков А. А., Морозов А. В., Прошин В. И.—Система проектирования программируемой логики 2
 Корчак А. Е.—Интерпретатор языка программирования Бейсик/F для микроЭВМ 3
 Крылов В. В.—Наука о программах (письмо в редакцию) 1
 Лавров С. С., Слисенко А. О., Цейтин Г. С.—Проект плана-программы по специальности «Информатика и системное программирование» 4
 Липаев В. В., Каганов Ф. А., Керданов А. В., Загубин Ю. В., Колдобский А. Э., Новичков А. С.—Система автоматизации проектирования программ на базе персональных ЭВМ (система ПРА) 4
 Лосейт А. Т., Кочегарова С. И.—Опыт применения Р-технологии при разработке программных средств на микроЭВМ 4
 Лукьянов Д. А.—Как написать кросс-транслятор с языка ассемблер 4
 Поливцев С. А., Мороговский Б. Н., Шарай З. И.—Драйвер параллельной связи микроЭВМ с дисплеем и устройством мозаичной печати 2
 Туманов А. А.—Система подготовки программ для микропроцессора КР5801К80 на базе микроЭВМ «Искра 226» 1
 Цыганков В. А., Бураков Е. А., Гусев А. В., Козырев А. В., Котлин В. Н., Фаас В. Г., Черепанов А. Б.—Система программирования и отладки микроЭВМ на базе измерительно-вычислительного комплекса 1
 Щелкунов Н. Н., Дианов А. П.—Программирование микросистем реального времени 4

КАК УЧИТЬ ПРОГРАММИРОВАНИЮ

Лосев И. С.—Размышления об обучении программированию 3

Варсанюфьев Д. В., Кушпиренко А. Г., Лебедев Г. В.—Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники 3

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Бамбуров Н. С., Екимов А. В., Ермолин Ю. В., Мамаков П. В., Сытин А. Н.—Программируемый групповой контроллер на базе МПК БИС серии К580 4

Белош В. В., Путилов В. А., Смольков Г. Я.—Система автоматизации Сибирского солнечного радиотелескопа 1

Блинов Н. А., Касьянов В. В., Паничев А. В.—Помехоустойчивый АЦП, сопрягаемый с микропроцессором КР580ИК80 2

Бураков Е. А.—Микропроцессорный кардиоинтерпретатор 1

Бужан Ю. Д., Осипов И. Н.—Панель управления микропроцессорной системой 4

Всронин В. И., Макаров К. В., Старшова В. А.—Контроллер клавиатуры на базе однокристалльной микроЭВМ КМ1816ВЕ48 3

Говоруи В. Н., Ермолин Ю. В., Мамаков П. В., Рыбаков В. Г., Сытин А. Н.—Микропроцессорные модули для автоматизированных установок физики высоких энергий 1

Горюнова С. В., Корсаков С. Я., Крылов В. В., Морозов А. В., Пономарев Д. М.—Система автоматизации производства на базе микроЭВМ «Электроника 60М» 2

Громов Г. Р.—Надежность персональных ЭВМ и производственная загрузка фирменной сети сервиса 4

Михайлов С. А.—Система управления электроприводами судовых грузовых комплексов 4

Петух А. М., Романюк А. Н., Подольский О. А.—Преобразователь интерфейсов ИРПС — «Общая шина» 4

Преснухин Л. Н., Фролов Г. И., Куправа Т. А., Безобразов В. С., Шахнов В. А.—Учебный класс на основе диалоговых вычислительных комплексов 3

Семенов П. А.—Микроконтроллеры на базе БИС КР580 для микроЭВМ «Электроника 60» и «Электроника НЦ-80-01Д» 3

Семенов Г. Н.—Автоматизированный комплекс на базе микроЭВМ «Электроника 60М» для управления бурением 4

Сумин В. В., Кислицын А. Б., Воробьев В. И., Савин А. Г.—Простейшие программируемые измерительные приборы на базе микроЭВМ 3

Тарасов Н. А.—Устройство обработки информации с малым энергопотреблением 3

Торгов Ю. И.—Программируемый 8-голосый синтезатор на базе микроЭВМ 2

Филиппычев С. А., Майдыковский И. В., Борщенко Ю. И., Зубов Ю. В.—Применение однокристалльного микропроцессора К1801ВМ1 в автономных системах сбора и обработки информации 1

ГИБКИЕ АВТОМАТИЗИРОВАННЫЕ ПРОИЗВОДСТВА

Адамов Е. О., Дукарский С. М., Корягин Д. А.—Интегрированная машиностроительная система КАПРИ 4

Адамов Е. О., Гримайло С. И., Камынин С. С., Корягин Д. А., Смольянов Ю. П.—Роботизированный механообрабатывающий комплекс для участка гибкого производства 4

Каляев А. В., Чернухин Ю. В., Брюхомицкий Ю. А., Каляев И. А., Носков В. П.—Построение однородных управляющих структур адаптивных автономных роботов 4

Савин А. И., Кузнецов М. Н., Уксусов А. С., Целяпин А. Н.—Структура системы управления ГАП 4

Соломенцев Ю. М., Сосонкин В. Л.—Оперативное управление гибкими производственными системами 4

Смирнов Ю. С.—Системы управления сервомеханизмами с шаговыми электродвигателями 4

УЧЕБНЫЙ ЦЕНТР

Барабанов А. Б., Турунов Н. Г.—Простой программатор 2

Барабанов А. Б., Турунов Н. Г.—Клавишное устройство ввода информации 3

Буенков А. М., Бурбин Н. В., Титчев Н. И.—Организация совмещенного ввода с перфолент в микроЭВМ «Электроника 60М» 3

Гребенников В. А.—Упрощенный пультовой терминал к микроЭВМ типа «Электроника 60М» 2

Грушвицкий Р. И., Коровацкий В. П., Преображенский А. В.—Особенности построения и использования учебных микроЭВМ 2

Зеленко Г. В.—Дисплей для бытовой персональной ЭВМ 3

Как оформлять программы для публикации 2

Корольков А. А., Раденко М. Е., Сеньков В. К.—Применение БИС КР580ВВ51 для реализации последовательных интерфейсов микропроцессорных систем 1

Крылов Е. Т., Крылова Е. Э., Медведева Л. П.—Быстродействующий аналого-цифровой преобразователь для микропроцессорных систем 1

Лукьянов Д. А.—КР580 — автоматизация без проблем! 1

Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г.—Устройство прямого доступа к памяти микроЭВМ 2

Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г.—Принципы организации и работы дисплеев на основе БИС КР580ВГ75 3

Преснухин Л. Н., Панфилов Д. И., Романенко О. А., Шаронин С. Г.—Лаборатория по изучению микропроцессорных комплексов с фиксированным набором команд 1

Торгов Ю. И.—Использование нескольких микропроцессоров в микроЭВМ 1

Широков Ю. Ф., Осипов Е. Н., Жданов В. И., Гутовец Н. И.—Учебно-отладочное устройство «Электроника 580» 2

УЧИТЬ РАБОТАТЬ С ПЗУ

Щербаков О. А.—Физические основы записи информации в ПЗУ 3

Дианов А. П., Щелкунов Н. Н.—Методика программирования микросхем ПЗУ 3

Дианов А. П., Щелкунов Н. Н.—Модуль программирования микросхем ПЗУ 3

Лукьянов Д. А.—Схемотехника универсальных программаторов ПЗУ 3

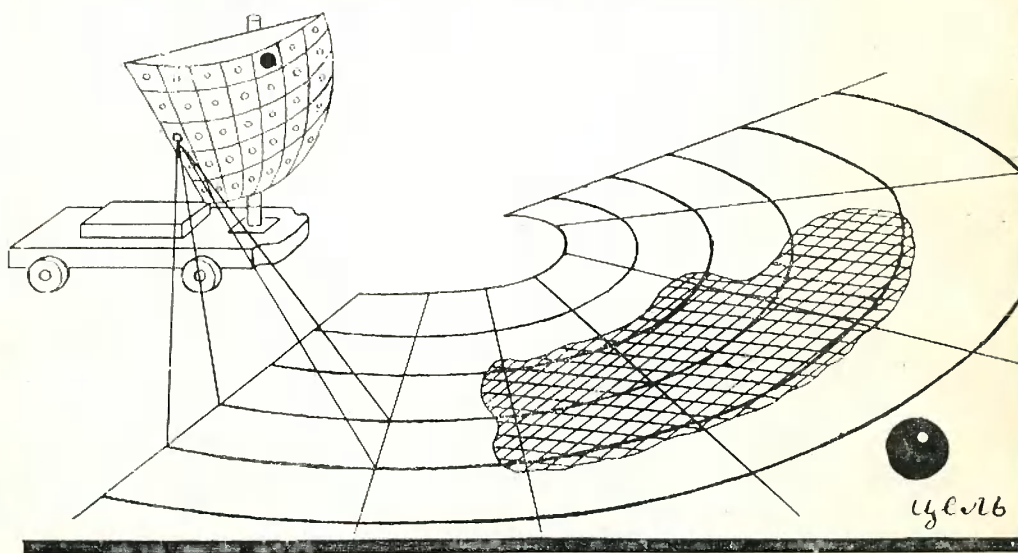
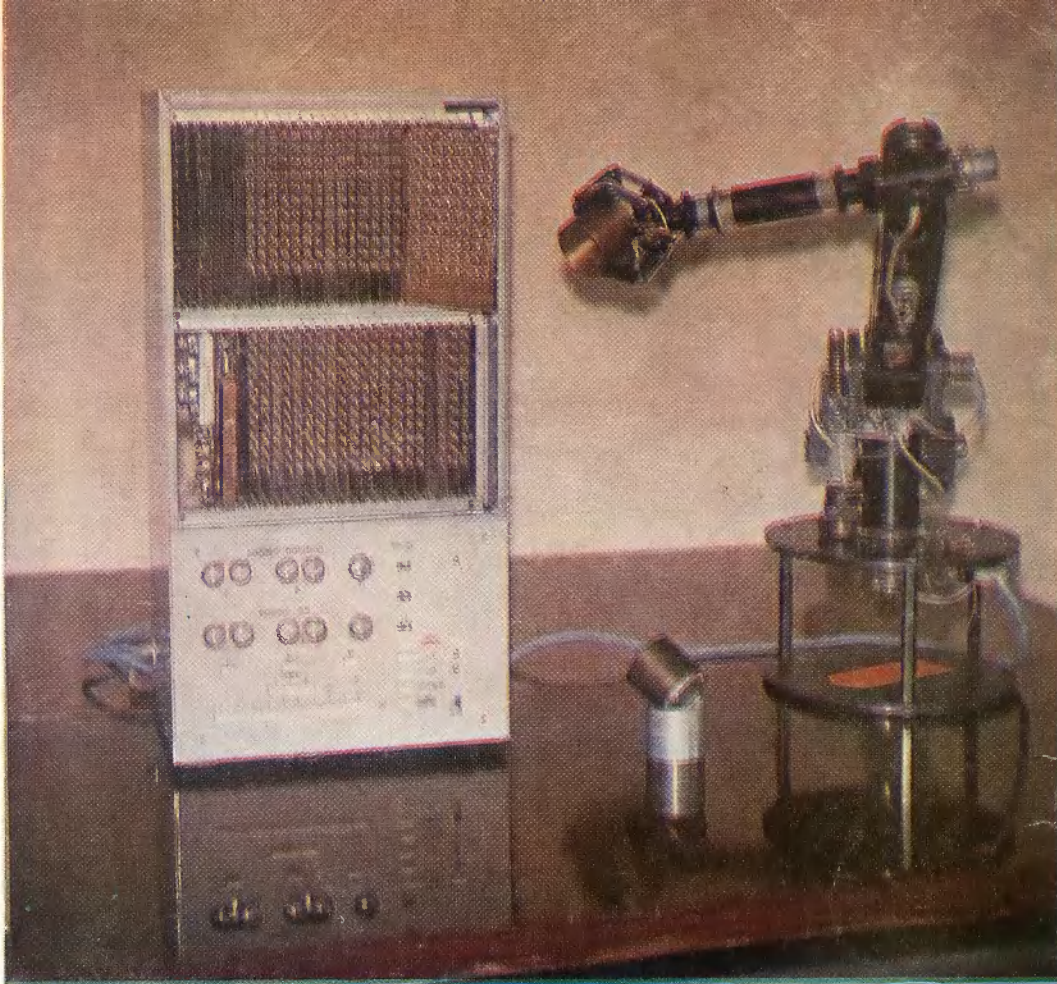
СПРАВОЧНАЯ ИНФОРМАЦИЯ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Малашевич Б. М., Шахнов В. А., Коночкин Э. И.—Внешние запоминающие устройства 1

Павлов В. В.—Головной консультационно-технический центр по применению микропроцессоров 2

Торгов Ю. И.—Ответ на письмо читателя 3

Использование БИС ОУС в системах управления адаптивными транспортными роботами позволит существенно улучшить их массо-габаритные характеристики, повысить надежность и живучесть, снизить стоимость. Применение только одной БИС ОУС позволит заменить комплект аппаратуры, содержащий 1500 микросхем средней степени интеграции.



Макет управляющей системы пятистепенного робота-манипулятора (фото вверху)
 Макет адаптивного транспортного робота (фото внизу)
 (к ст. Каляева А. В. и др. «Построение однородных управляющих структур адаптивных автономных роботов»)

Сушикову

1р. 10к.

Индекс 70588

ВНИМАНИЕ

Если Вы не успели подписаться на наш журнал, НАПОМИНАЕМ, что в 1986 году подписка принимается ПОКВАРТАЛЬНО. За месяц до начала любого квартала прекращается подписка на ту часть годового комплекта журналов, которая издается в следующем трехмесячном периоде. Таким образом, до начала марта Вы можете подписаться на журналы, начиная с 3-го номера и т. д.

ВНИМАНИЕ

С 1986 ГОДА ЖУРНАЛ ВЫХОДИТ 6 (ШЕСТЬ!) РАЗ В ГОД!

Если Вы подписались только на 4 номера, то имеется возможность оформить дополнительную подписку на 5 и 6 номера годового комплекта. Сделать это Вы можете в течение первого полугодия 1986 года в любом отделении «Союзпечать».

ВНИМАНИЕ

С 1986 года журнал издается 6 (шесть!) раз в год!

Подписка на журнал ежеквартальная и принимается всеми отделениями Союзпечати в течение всего года!

Цена одного номера — 1 руб. 10 коп. Индекс журнала — 70588