



МИКРО- ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

5-6 / 1989

ISSN 0233-4844

Программируемый микроконтроллер на основе ОЭВМ КР1816ВЕ31 управляет технологическими объектами, «интеллектуальными» приборами, бытовой техникой

Тестовый процессор на базе микросхем серии К1810 обеспечивает селективный контроль информации, передаваемой по распределенной магистрали микропроцессорных систем

Мультимикропроцессорная система спектрального анализа сигналов в реальном масштабе времени: основные принципы построения системы обработки аналогового сигнала
Простой аппаратно-программный интерфейс между ПЭВМ ЕС1840 и ДВКЗ на основе последовательных интерфейсов RS232 и ИРПС

Одноплатное ОЗУ с контролем хранения данных повышает надежность вычислительных систем

Кронос: семейство универсальных процессоров с аппаратной поддержкой языков высокого уровня для конструирования ЭВМ открытой архитектуры

Комплексная автоматизированная система отладки 8-, 16-, 32-разрядных микропроцессоров MSW16/32 — обеспечивает проектирование аппаратных и программных средств с минимальными затратами

Инструментальные средства разработки экспертных систем на ПЭВМ — отечественный и зарубежный опыт



ЛАБОРАТОРИЯ МП ТЕХНИКИ С ЛОКАЛЬНОЙ СЕТЬЮ ПЭВМ «ЭЛЕКТРОНИКА БК-0010»

На кафедре электроники и микропроцессорной техники Московского лесотехнического института разработан программно-технический комплекс, включающий лабораторные стенды с микромакетами, локальную сеть ДВК2М и ПЭВМ «Электроника БК-0010» (далее БК), сервисные и обучающие программы и инструментальную систему для их разработки.

Лабораторные стенды (см. рис.) оснащены съемными микромакетами МП К580ИК80, К1816ВЕ48 со схемами тактовых генераторов и пошагового режима работы, БИС ОЗУ, регистров К589ИР12, параллельного интерфейса К580ВВ55, таймера К580ВИ53, контроллера прерываний К580ВН59, последовательного интерфейса К580ВВ51, а также набора ИС серии 155. Микромакет соединены проводниками и плоскими кабелями. В средней части стенда расположены 16-ричный индикатор данных и разъем параллельного интерфейса с БК.

Локальная сеть состоит из ДВК2М и шести БК. Сеть выполняет единственную функцию загрузки программ: из ДВК в БК. Для этого на ДВК реализована программа, имитирующая на одном из разрядов выходного порта платы И2 выход магнитофона в режиме считывания файла с кассеты в БК. Используя способность драйвера магнитофона БК настраиваться на частоту приема, достигнута скорость передачи 3600 бод. Вместо платы И2 можно использовать любой контроллер параллельного интерфейса, в том числе и порт печатающего устройства ДВК.

В комплект сервисных и обучающих программ БК входят кроссассемблеры для МП К580ИК80 и К1816ВЕ48, многоканальный цифровой анализатор и пополняемый набор обучающих и сопровождающих лабораторные работы программы.

Кроссассемблеры содержат встроенный редактор, обеспечивающий ввод текстов программ с клавиатуры или с магнитофона, редактирование и запись программ на магнитофон. Программы ассемблируются и загружаются в ОЗУ сматерированной на стенде МПС. После этого возможны: выполнение программы с выводом результатов на 16-ричный индикатор, отладка в пошаговом режиме и модификация программы непосредственно в ОЗУ МПС. Кроссассемблеры ориентированы исключительно на учебные цели и из-за малого объема оперативной памяти БК имеют ряд ограничений: длина строки — до 20 символов,

число строк — до 100, объем результирующего кода — до 256 байт.

Программа многоканального цифрового анализатора предназначена для ввода 16 цифровых сигналов через параллельный порт БК, их запоминания и отображения в виде временных диаграмм. В отличие от типовых промышленных цифровых анализаторов в программе есть возможность присвоения имен сигналам, что существенно облегчает чтение и анализ диаграмм, установки состояния сигналов, при котором начинается их сканирование, выбора окна просмотра и масштаба. Поскольку анализатор реализован программно, его рабочая частота ограничена несколькими килогерцами. Поэтому частота работы МП в микромакетах соответственно снижена.

Сопровождающие и обучающие программы контролируют ход выполнения лабораторных работ, выдают справочную информацию в текстовом и графическом виде, в том числе схемы МПС, проверяют знания студентов и формируют необходимую информацию для преподавателя.

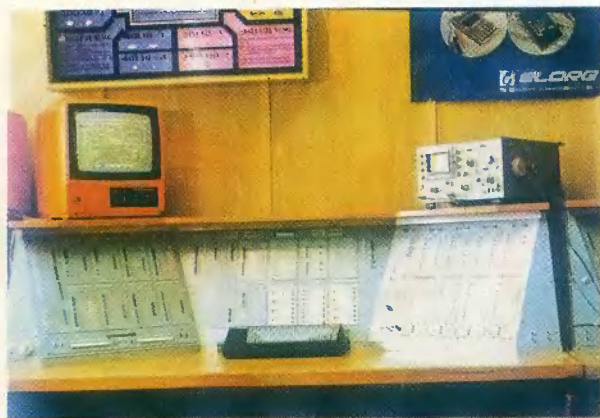
Инструментальная система для разработки программно обеспечения для БК состоит из кросс-системы программирования на языке Си и генератора диалогов. Кросс-система позволяет разрабатывать и отлаживать программы для БК на Си в среде ОС РАФОС на ДВК. Для этого создана библиотека функций работы с экраном и клавиатурой БК в стандарте Си, что позволяет переносить диалоговые программы с ДВК на БК без изменений. В эту библиотеку также включены функции, реализующие расширенные возможности БК в части графики, цвета и звука.

Отладку программ можно производить автономно на ДВК и в комплексе с БК. В последнем случае БК с соответствующей программой выполняет роль цветного графического терминала, а ДВК — процессора с монитором диалоговой отладки. В режиме комплексной отладки БК связан с ДВК через 16-разрядный параллельный интерфейс с программно реализованным асинхронным протоколом, обеспечивающим скорость обмена около 7 Кбайт/с. Драйвер параллельного интерфейса загружается в БК либо с магнитофона, либо по локальной сети, а следующие программы, в том числе и эмулятор графического терминала, — по параллельному интерфейсу.

Язык Си в отличие от Паскаля дает заметный выигрыш как в памяти, так и в быстродействии за счет возможности использования регистровых переменных, инициализации констант на этапе трансляции, гибкой работы с адресами и, как правило, генерации более эффективного кода.

Генератор диалогов предназначен для разработки и отладки программ сопровождения лабораторных работ без программистов. Для этого диалог со студентом, выполняющим лабораторную работу, проектируется в виде ориентированного взвешенного графа. Вершины графа соответствуют кадрам диалога, дуги — ветвям диалога, а веса — баллам, оценивающим ответ студента на вопрос, заданный в кадре диалога. Помимо кадров — вопросов в диалоге можно использовать следующие типы кадров: информационный, транзитный — для схода вопроса, на который ранее был дан верный ответ, анализа — для выбора ветви диалога в соответствии с суммой набранных баллов, вставки — для повторного использования тексто-графических изображений в разных кадрах.

Телефон: 582-45-34, Чернышов Юрий Николаевич



Лабораторный стенд

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года

МГ МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 5 /1989 МОСКВА

СОДЕРЖАНИЕ

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Машинная графика

ЯП Паскаль — вопросы реализации

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Медицинские приложения

Микроконтроллеры

Периферийные устройства и УСО В блокнот разработчика Справочная информация

Кобылинский А. В., Калатинец В. М., Заика А. И.— Микропроцессор КМ1810ВМ87 с плавающей запятой	2
Антонюк Б. В., Омельчук И. В., Присяжнюк В. Н., Терещенко Г. Г.— Тестовый процессор для мультипроцессорных систем	13
Розенштейн Э. П., Овчинников А. В.— Синхронизация работы микропроцессоров серии К1801	17
Полянский П. В.— Организация прикладных баз данных на ПЭВМ с малой емкостью ОЗУ	20
Казменко С. В.— База данных при интеграции диалоговых средств конечного пользователя	23
Корытный И. М., Мукалов И. О.— Объектно-ориентированная реализация диалога при доступе к реляционной базе данных	28
Авдеев В. Н., Белов А. Е., Жедь А. Ю., Корнюшко В. Ф.— Интерпретатор языка БЕЙСИК со встроенным графическим пакетом для ДВК	31
Богатырев В. А.— Арбитр с децентрализованным кодовым управлением	33
Коллегов М. В., Репков В. В.— Дисплей с высоким разрешением	35
Данилов Н. Ю., Полукаров А. Д., Столяров М. В.— Дисплейные микроконтроллеры семейства МИДИКОН	38
Лацис А. О., Штаркман В. С.— Системы команд растрового графического дисплея «Электроника МС7401»	39
Бучнев А. А., Сизых В. Г., Минин В. Ф.— Цветная полутоновая станция профессионального назначения ГАММА-7.1	45
Васьков С. Т., Коломеев О. А., Перекрест О. В., Ситников Г. Ф., Ткач С. Е.— Цветной монитор с высоким разрешением для растровых дисплеев	46
Добриневский С. Ф.— Процедуры обработки прерывания в Паскале ОС ДВК	49
Грибов И. В., Зиновьев С. В., Шумаков А. В.— Программирование систем реального времени на языке Паскаль	52
Горшенин А. Г., Воронов В. Б.— Звуковое сопровождение программ в Паскале М86 для ПЭВМ ЕС1840	55
Зайцев В. А., Кротов С. Н., Прошкин С. Д., Савченко А. В.— Мультимикропроцессорная система спектрального анализа сигналов в реальном масштабе времени	57
Падиряков Ю. А., Белинский В. Т., Журило В. А.— Модуль системы обработки данных реального времени	62
Мифтахов М. Г., Бортасевич В. С., Мифтахова Г. Р.— Многопроцессорная информационно-управляющая система с каналом ПДП для геофизических исследований скважин	64
Иванов А. П., Осьмаков А. Н., Лысенко В. В.— Система для сбора, обработки и регистрации информации на кассетной МЛ	66
Лукьянов В. И., Логинов О. Е., Даньков В. Б., Тетерина Е. А., Бородин С. М., Блинов С. М., Никандров М. Г.— Информационно-измерительная система для исследования времени двигательных реакций человека	68
Вайнштейн Л. Г.— Вычислительный комплекс для обработки электрофизиологической информации в условиях нейрохирургического стационара	70
Захаров С. В., Марфенко К. С., Подлепечий Б. И., Торубаров С. В.— Система на базе МПК БИС КР580 и КМ1813ВЕ1 для мониторинга физиологических параметров человека	74
Щелкунов Н. Н., Дианов А. П.— Организация однокристалльных микроконтроллеров	76
Орестов Ю. А., Бобылев В. Н.— Программируемый микроконтроллер на основе ОЭВМ КР1816ВЕ31	83
Андрюшин О. С.— Сопряжение ДВК с устройством ввода графической информации типа СМП6410	86
Гринь Н. Ф., Уткина Т. Г.— Модем связи с бытовым магнитофоном	88
Сунгуров В. С., Ионов Н. Н.— Доработки БИС серии КР588	91
Фейзханов У. Ф.— Ошибка в операционной системе для ДВКЗ	92
Постоянное запоминающее устройство КР1801РЕ2	94

МИКРОПРОЦЕССОР КМ1810ВМ87 С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Общая характеристика микросхемы

СБИС КМ1810ВМ87 — однокристалльный специализированный микропроцессор (сопроцессор), предназначенный для выполнения числовых операций с высокой производительностью и точностью в формате с фиксированной и плавающей запятой. Данный микропроцессор (МП) существенно расширяет функциональные возможности мультипроцессорных систем на основе центрального процессора (ЦП) КМ1810ВМ86 или КМ1810ВМ88, который при этом должен работать в максимальном режиме*. Комбинация МП КМ1810ВМ87 с ЦП КМ1810ВМ86 (КМ1810ВМ88) воспринимается программистом как единый процессор с расширенными системой команд и типами данных. Программные средства

и внутренняя архитектура МП КМ1810ВМ87 позволяют автоматически синхронизировать его с ЦП при их совместной работе.

В систему команд (их 69) МП КМ1810ВМ87 входят 16 команд управления, 9 — пересылки данных, 25 — арифметических, 7 — сравнения, 5 — трансцендентных функций, 7 — загрузки констант. МП КМ1810ВМ87 производит операции с семью типами данных, включающих три формата двоичных целых чисел длиной 16, 32 и 64 двоичных разряда; один формат двоично-десятичных целых чисел длиной 80 двоичных разрядов; три формата действительных чисел с плавающей запятой длиной 32, 64 и 80 разрядов. МП позволяет также представлять специальные величины: плюс ноль, минус ноль, плюс бесконечность, минус бесконечность,

неопределенность (замаскированный ответ на особый случай — недействительная операция), нечисловую величину, денормализованные числа. При выполнении команд МП КМ1810ВМ87 контролирует особые случаи наличия денормализованного операнда недействительной операции, переполнения, антипереполнения, деления на ноль, неточности. При обнаружении немаскируемых особых случаев МП приостанавливает выполнение программы, о чем сообщает ЦП.

Благодаря аппаратному способу реализации вычислений, в том числе операций умножения, деления, извлечения квадратного корня, тригонометрических тангенса и арктангенса, МП КМ1810ВМ87 повышает производительность микропроцессорной системы при вычислении сложных процедур на два и более порядка. Внутренние операции в МП КМ1810ВМ87 производятся в диапазоне действительных чисел $\pm 3,4 \times 10^{-4932} \dots \pm 1,2 \cdot 10^{4932}$, а вычисления — с перепрограммируемой точностью 64, 53 или 24 двоичных разряда.

Обращение к сопроцессору осуществляется только по его командам (нет необходимости в специальных командах адресации сопроцессора). В памяти числовая информация может быть представлена 2-, 4- и 8-байтными целочисленными данными, 10-байтными упакованными двоично-десятичными данными, 4-, 8- и 10-байтными действительными данными. Конструктивно микросхема выполнена в металлокерамическом 40-выводном корпусе типа 2123.40—9.01 с двухрядным расположением выводов (рис. 1). Условное графическое изображение микросхемы показано на рис. 2. Назначение выводов микросхемы приведено в табл. 1.

Структурная организация микропроцессора

МП КМ1810ВМ87 состоит из двух функциональных частей — устройства управления и исполнительного устройства (рис. 3). Устройство управления синхронизирует работу ЦП и сопроцессора. Основные узлы его: регистры управляющего слова (РУС), слова состояния (РСС), указателя текущей команды (РУТК), предварительной очереди данных (РПОД) и модуль синхронизации (МС). Исполнительное устройство выполняет все операции по обработке данных. Основные узлы его: регистровый стек (восемь 80-разрядных регистров), микропрограммный модуль (МКПМ), арифметический модуль



Рис. 1. Расположение выводов КМ1810ВМ87

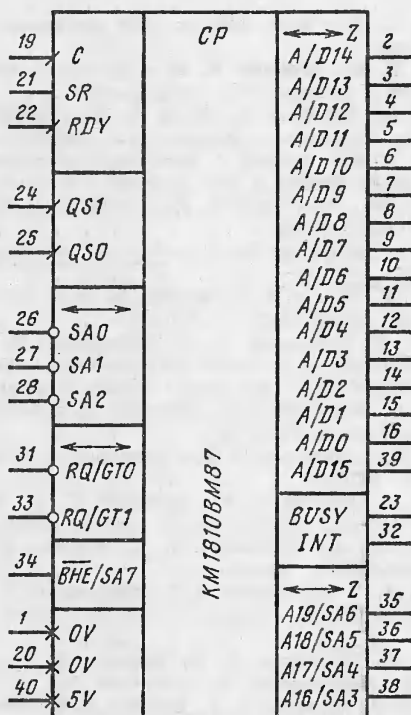


Рис. 2. Условное графическое изображение КМ1810ВМ87

* Кобылинский А. В., Москалевский А. И., Темченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86 // Микропроцессорные средства и системы. — 1986. — № 1. — С. 28—34.

Назначение выводов микросхемы КМ1810ВМ87

Вывод	Обозначение	Тип	Назначение
1,20	OV	—	Общие выводы
2...16,39	A/D14...A/D0, A/D15	Входы-выходы	Шина адреса/данных
17, 18, 29, 30	NC	—	Не используются
19	C	Вход	Синхронизация
21	SR	»	Установка (сброс)
22	RDY	»	Готовность
23	BUSY	Выход	Занят
24, 25	QS1, QS0	Входы	Сигналы выборки команд из очереди команд
26...28	SA2...SA0	Входы-выходы	Состояние цикла шины
31, 33	RQ/GT0, RQ/GT1	»	Запрос/разрешение на владение шиной A/15...A/0
32	INT	Выход	Запрос прерывания
34	BHE/SA7	Вход-выход	Разрешение передачи по старшей половине шины D15...D8/Шина адреса состояний
35...38	A19/SA6...A16/SA3	Входы-выходы	Шина адреса состояний
40	U _{cc}	—	5 В

(AM), модуль обработки порядка (МОП), модуль программного сдвига (МПС), регистры временного хранения (РВХ) и регистр слова признаков (РСР).

Управляющее слово, слово состояния, слово признаков и указатель текущей команды — частичный набор программно-доступных средств (ПДС) сопроцессора, который вместе с регистровым стеком образуют полный набор ПДС. Содержимое ПДС может быть

сохранено (запомнено) в памяти или восстановлено (загружено) по содержанию памяти с помощью команд сопроцессора.

Устройство управления

Управляющее слово (рис. 4) обеспечивает широкий диапазон применения сопроцессора путем задания нескольких режимов его работы, которые устанавливаются по сигналу системного сброса или по одной из команд:

«Загрузка управляющего слова» (FLDCW), «Загрузка частичного набора ПДС» (FLDENV), «Восстановление содержимого полного набора ПДС» (FRSTOR), «Начальная установка» (FINIT/FNINIT).

Поля масок особых случаев IM, DM, ZM, OM, UM, PM (разряды D0...D5) определяют способ обработки соответствующего случая. Если маска установлена (состояние «1»), то используется встроенная процедура обработки особого случая, в противном случае — процедура обработки особого случая, определенная пользователем при условии, что прерывания не замаскированы.

Поле маски запроса на прерывание IEM разрешает (маскирует состояние «0») или запрещает (маскирует состояние «1») формирование запроса на прерывание при возникновении особого случая (не маскированного). По прерыванию управление передается процедуре обработки особого случая, определенной пользователем. Запрещение запроса на прерывание эквивалентно маскированию всех особых случаев.

Поле управления точностью PC (разряды 8 и 9) изменяет точность представления данных во внутреннем (10-байтном действительном) формате. В зависимости от содержимого поля PC число значащих разрядов в мантиссе может быть уменьшено до 53 или 24 (размер поля порядка при этом не меняется):

Содержимое поля PC	Размер мантиссы, двоичных разрядов
0	24
1	24
2	53
3	64

(соответствует 10-байтовому внутреннему формату)

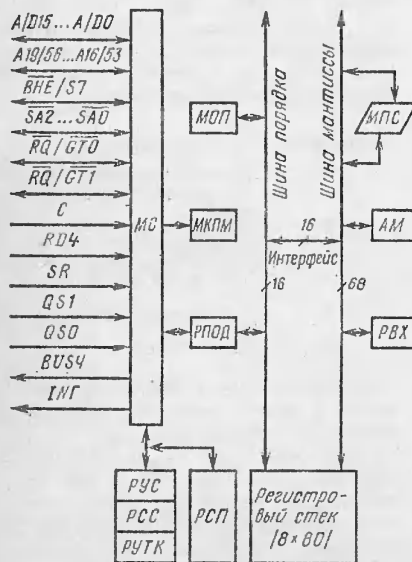


Рис. 3. Структурная организация КМ1810ВМ87

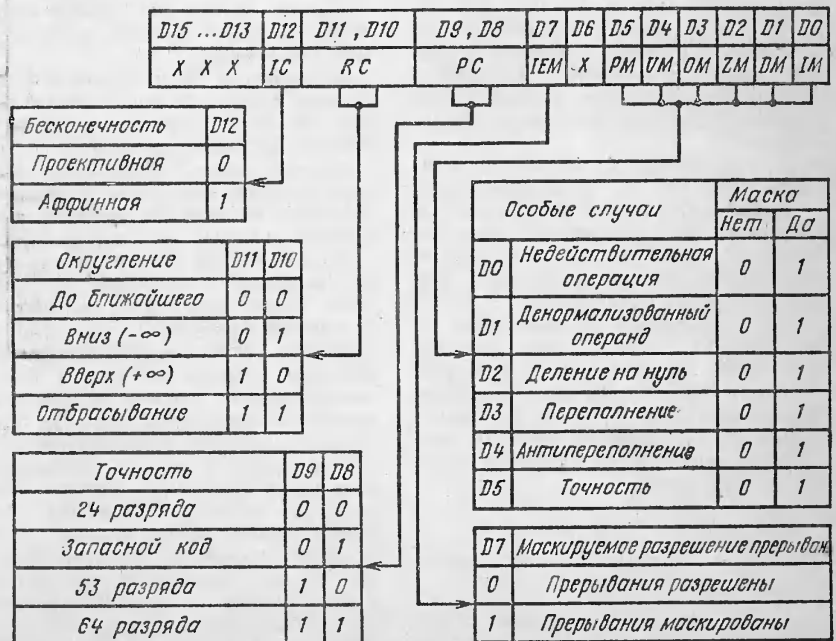


Рис. 4. Формат управляющего слова

Уменьшение числа значащих разрядов в мантиссе внутреннего формата представления данных не приводит к увеличению скорости выполнения операций сопроцессором.

Результат округляется до указанного размера мантиссы в соответствии с полем управления округлением RC. Неиспользуемые разряды находятся в состоянии «0». Это поле устанавливает один из четырех режимов округления результата выполнения операций сопроцессора:

Содержимое поля RC	Режим округления до ближайшего числа
0	Или четного
1	Меньшего
2	Большого
3	По абсолютной величине

Если результат выполнения операции В не может быть размещен в определенном командой формате и находится между двумя ближайшими представимыми числами А и С, т. е. $A < B < C$, то он округляется в соответствии с указанным режимом. Округление вносит ошибку, не превышающую единицы последнего разряда числа, до которого производится округление. При округлении:

до ближайшего или четного числа выбирается ближайшее из чисел А или С. Если число В находится посередине между числами А и С, то выбирается четное число, младший значащий разряд которого равен нулю. Данный режим обеспечивает наиболее точную и статически уравновешенную ошибку;

до ближайшего меньшего числа (т. е. в сторону отрицательной бесконечности) и до ближайшего большего числа (т. е. в сторону положительной бесконечности) в качестве результата В принимается число А. Эти режимы можно использовать для вычисления верхнего и нижнего пределов результата путем двухразового выполнения программы вычисления с применением различных режимов округления результата;

до ближайшего по абсолютной величине числа (т. е. в сторону нуля) в качестве результата В принимается меньшее по абсолютной величине число А или С. Данный режим округления обычно применяется при целочисленных вычислениях.

Поле управления бесконечностью IC устанавливает один из двух режимов обработки бесконечности: с использованием (аффинная) и без использования (проективная) знака. В зависимости от содержимого поля IC вид используемой в вычислениях бесконечности следующий:

Содержимое поля IC	Вид бесконечности
0	Проективная (без знака)
1	Аффинная (со знаком)

Слово состояния отражает общее состояние сопроцессора (рис. 5). Поля флагов (IE, DE, ZE, OE, UE, PE) указывают на наличие или отсутствие

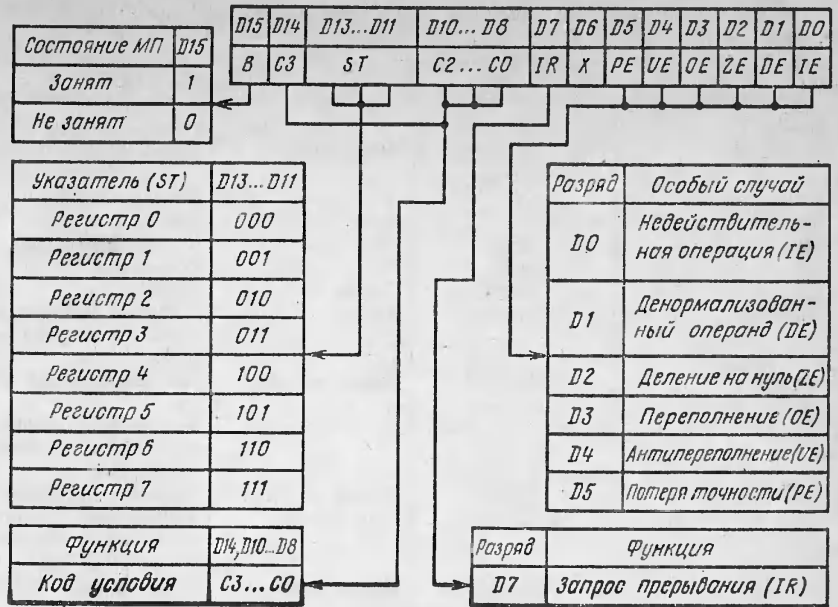


Рис. 5. Формат регистра слова состояния

особых случаев, возникающих при выполнении текущей команды сопроцессора. Если особый случай имел место, то поле соответствующего флага устанавливается в состояние «1», в противном случае — в состояние «0». Сущность особых случаев в следующем:

недействительная операция (IE) возникает при невозможности выполнения операции сопроцессором;

денормализованный операнд (DE) — при попытке выполнения операции над денормализованным операндом;

деление на ноль (ZE) — при выполнении операции деления, когда делитель равен нулю;

переполнение (OE) — если величина порядка результата выполнения операции выходит за пределы (становится больше) его представления;

антипереполнение (UE) — если величина порядка результата выполнения операции выходит за пределы (становится меньше) его представления; *потеря точности (PE)* — если результат выполнения операции не может быть точно представлен в формате устройства примемника.

Особые случаи — *недействительная операция, деление на ноль и денормализованный операнд* — обнаруживаются до выполнения операции и называются ранними; *переполнение, антипереполнение и потеря точности* — в процессе выполнения операции и называются поздними. Если был обнаружен ранний особый случай, то содержимое устройства-примемника команды (т. е. содержимое регистрового стека и памяти) не изменяется (команда как бы не выполнялась). При обнаружении позднего особого случая содержимое этого устройства может быть изменено.

Незамаскированные поздние особые случаи *переполнение* и *антипереполнение* обрабатываются, как ранние особые случаи (память не обновляется, выборки из стека не происходит), при выполнении команд «Запоминание целочисленного данного» (FIST), «Запоминание целочисленного данного и выгрузка содержимого текущего регистра из стека» (FISTP), «Запоминание вещественного данного» (FST) и «Запоминание вещественного данного и выгрузка содержимого текущего регистра из стека» (FSTP). Кроме того, может возникнуть несколько особых случаев, однако устанавливается флаг только одного из них в соответствии с последовательностью приоритета:

Приоритет	Особый случай
1	Денормализованный операнд (если особый случай не замаскирован)
2	Недействительная операция
3	Деление на ноль
4	Денормализованный операнд (если особый случай замаскирован)
5	Переполнение и антипереполнение
6	Потеря точности

При возникновении особого случая его флаг в слове состояния устанавливается в состояние «1».

Если маска особого случая (поля IM, DM, ZM, OM, UM и PM в управляющем слове) или маска запроса на прерывание (поле IEM управляющего слова) установлена в состояние «1», то используется встроенная процедура обработки особого случая, выполняемая сопроцессором; если маски особого случая и запроса на прерывание установлены в состояние «0», то используется внешняя (определенная

пользователем процедура обработки особого случая.

Вызов и выполнение встроенной процедуры называются маскированным ответом сопроцессора; вызов внешней (определенной пользователем) процедуры обработки особого случая — немаскированным ответом сопроцессора на особый случай. По маскированному ответу вырабатывается приближенный или наиболее вероятный результат той операции, при выполнении которой был обнаружен особый случай. Затем процесс вычисления продолжается.

По немаскированному ответу на особый случай формируется запрос на прерывание на выводе INT сопроцессора. При этом поле запроса на прерывание IR слова состояния устанавливается в состояние «1». Сопроцессор переходит в состояние ожидания (поле занятости сопроцессора в слове состояния находится в состоянии «1», т. е. сопроцессор занят выполнением операции) до момента установки в состояние «0» поля IR одной из команд «Установка в нуль полей в слове состояния» (FNCLEX), «Сохранение содержимого полного набора ПДС» (FNSAVE) или «Начальная установка» (FNINIT). После этого выполнение операции, вызывающей особый случай, продолжается.

В случае, если сигнал запроса на прерывание маскирован в контроллере прерываний (триггер IMR — в состоянии «1») или прерывания ЦП запрещены (триггер IF — в состоянии «0»), существует опасность возникновения состояния бесконечного ожидания, когда ЦП, выполняя команду «Ожидание» (WAIT), будет ждать сигнала освобождения сопроцессора на выводе TEST (BUSY), а сопроцессор — установки в состояние «0» поля IEM управляющего слова (рис. 6). Если особый случай маскирован, то при выполнении одной команды возникнет еще один или несколько особых случаев, так как процесс выполнения команды продолжается после маскируемого ответа на этот случай. Пользователю рекомендуется маскировать все особые случаи, за исключением случая *недействительной операции*, который обрабатывают, как фатальную ошибку в программе.



Рис. 6. Типичный путь прохождения сигнала запроса на прерывание

Определенная пользователем процедура обработки особого случая может состоять из сохранения состояния полного или частичного набора программно-доступных устройств сопроцессора, установки и состояния «0» содержимого флагов особых случаев в слове состояния (снятие запроса на прерывание), разрешения прерывания ЦП, идентификации и обработки особых случаев, возобновления нормального выполнения программы.

Процедура обработки особого случая должна устанавливать поле IR в состояние «0» перед возобновлением выполнения основной программы (перед возвратом в основную программу). В противном случае эта процедура может быть вызвана повторно по сигналу INT на выводе сопроцессора (т. е. может возникнуть состояние бесконечного ожидания). Если в процедуре обслуживания прерывания ЦП используются команды сопроцессора, то возврат в прерванную программу возможен только после завершения текущей команды сопроцессора. Для этой цели в программе перед возвратом из процедуры должна следовать команда «Ожидание ЦП, пока сопроцессор занят» (FWAIT).

Поле IR (разряд 7) отражает наличие запроса на прерывание при возникновении особого случая. Поле ST (разряды 11 ... 13) определяет адрес текущего регистра стека.

Поле битов условия CO, C1, C2, C3 (разряды 8 ... 10 и 14) характеризует результат выполнения таких команд, как «Сравнение вещественных данных» (FCOM), «Сравнение вещественных данных и выгрузка из стека» (FCOMP), «Сравнение вещественных данных и выгрузка из стека текущего и регистра 1» (FCOMPP), «Сравнение с целочисленными данными и выгрузка содержимого текущего регистра из стека» (FICOMP), «Сравнение содержимого текущего регистра с положительным вещественным нулем» (FTST), «Вычисление частичного остатка» (FPREM), «Установка битов условий по содержанию текущего регистра стека» (FXAM).

Поле занятости сопроцессора B (разряд 15) указывает, что сопроцессор занят выполнением команды (поле — в состоянии «1») или находится в состоянии ожидания (поле — в состоянии «0»). Состояние поля B отражается на выводе BUSY.

Указатель текущей команды (рис. 7) позволяет определить физический адрес и код текущей команды сопроцессора, а также адрес ее операнда в памяти. Исключением являются команды управления. Содержимое указателя текущей команды можно использовать в процедуре обработки особых случаев для получения информации о команде, вызвавшей особый случай.

Модуль MC синхронизирует работу ЦП и сопроцессора в асинхронном и синхронном режимах. Так как время выполнения команд сопроцессора ESC

D19...D11	D10...D0
Адрес операнда	
—	Код операции команды
Адрес команды	

Рис. 7. Формат указателя текущей команды

ЦП значительно меньше времени выполнения этой команды сопроцессором, то ЦП вместо ожидания может реализовать собственные команды. Этот режим работы в вычислительной системе является асинхронным. Однако существуют два случая, когда требуется синхронизировать работу ЦП и сопроцессора:

1. Исполнительное устройство не может приступить к выполнению следующей команды до завершения предыдущей. Синхронизация в данном случае осуществляется с помощью команды WAIT ЦП, применяемой перед командой ESC.

2. Центральный сопроцессор должен использовать операнд команды сопроцессора из памяти вычислительной системы только после завершения выполнения команды с таким операндом. Данный вид синхронизации обеспечивается путем записи в рабочей программе команды сопроцессора FWAIT перед командой ЦП, которая обращается к операнду в памяти.

Модуль синхронизации также выполняет все команды управления, за исключением FSAVE/FNSAVE и FRSTOR. При этом MC может выполнять команду управления сопроцессора в то время, когда исполнительное устройство сопроцессора выполняет вычислительную команду.

Регистры предварительной очереди данных состоят из двух 16-разрядных двоичных регистров (или четырех 8-разрядных). Они обеспечивают согласование длины форматов данных в зависимости от типа ЦП и по четному или нечетному адресу расположены в памяти.

Исполнительное устройство

Регистровый стек (рис. 8) состоит из восьми 80-разрядных регистров, используемых в качестве собственного стека и (или) набора рабочих регистров. Адреса регистров стека находятся в пределах 0...7. Адрес верхнего регистра стека, называемого текущим, хранится в регистре указателя стека ST. Указатель стека размещается в слове состояния. Адрес текущего регистра (т. е. содержимое ST) может быть увеличен или уменьшен командами сопроцессора. При увеличении содержимого ST переходит из состояния «7» в состояние «0»; при уменьшении — из состояния «0» в состояние «7».

Формат регистра стека состоит из четырех полей: разряды 0 ... 62 — дробная

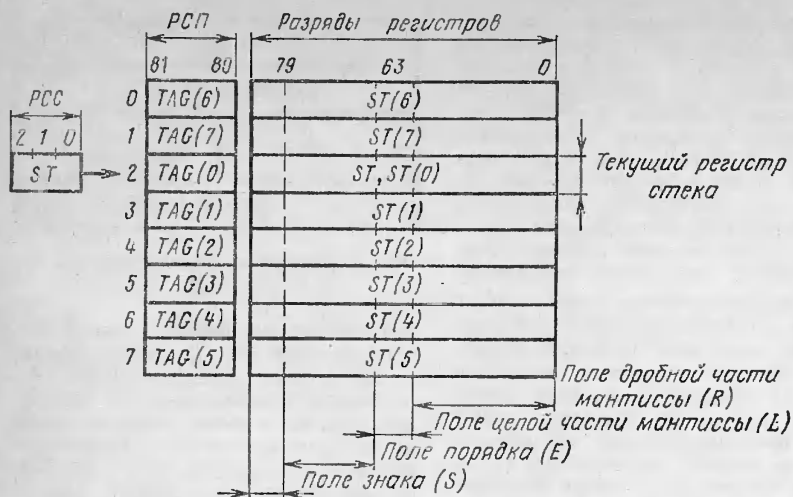


Рис. 8. Регистровый стек

часть мантииссы R; разряд 63 — целая часть мантииссы L; разряды 64...78 — порядок E; разряд 79 — знак S. Распределение полей соответствует внутреннему 10-байтному формату представления действительных данных.

В командах языка ассемблера используются явные и неявные ссылки на регистр стека. При неявной ссылке в формате команды не указывается имя регистра стека; при явной — указывается имя ST; при ссылке на произвольный ST — имя регистра стека с индексом ST (i), где i — абсолютное числовое выражение языка ассемблера со значением в пределах 0...7, представляющее индекс регистра относительно текущего ST.

Сопроцессор выполняет две операции: проталкивание в стек и выталкивание из него. При операции проталкивания (рис. 9) содержимое ST уменьшается на единицу (при этом изменяется положение текущего ST), данные записываются (загружаются) в текущий ST, по содержимому которого устанавливаются разряды соответ-

ствующего тега (признака содержимого) согласно табл. 2. При операции выталкивания из стека (рис. 10) данные считываются (выгружаются) из текущего ST, разряды его тега устанавливаются в состояние «Пусто», содержимое ST увеличивается на единицу (при этом изменяется положение текущего ST).

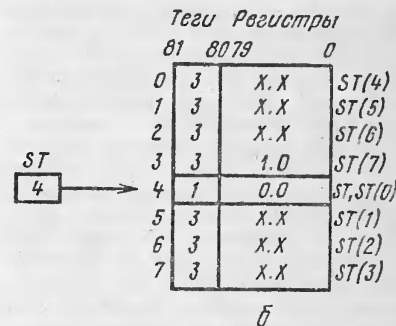
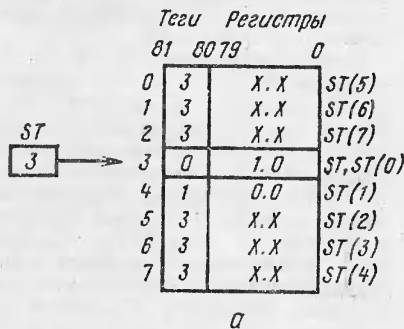


Рис. 10. Пример выполнения операции выталкивания из стека. Состояние стека до выполнения операции (а) и после (б)

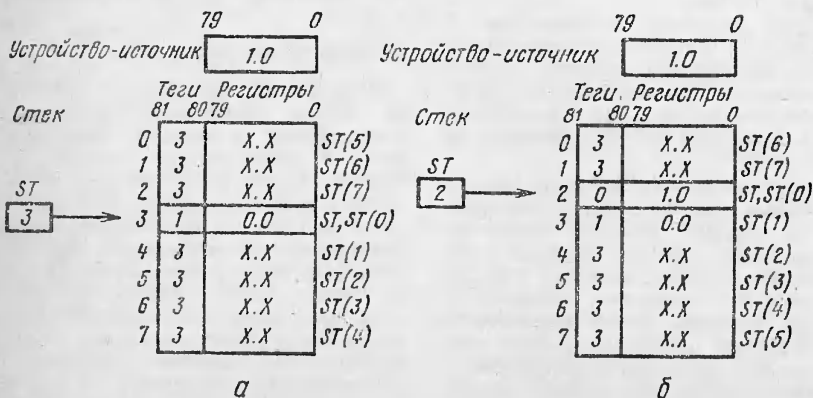


Рис. 9. Пример выполнения операции проталкивания в стек. Состояние стека до выполнения операции (а) и после (б)

Таблица 2
Регистр признаков

Содержимое регистра	Содержимое тега регистра
Число	
нормализованное	0
ненормализованное	0
Псевдоноль	1
Истинный нуль	1
Денормализованное число	2
Бесконечность	2
Не число (неопределенность)	3
Пустое	3

Содержимое регистра признаков (рис. 11) характеризуется его тегом, хранящимся в двух дополнительных разрядах (80 и 81) регистра стека (см. табл. 2). Разряды тегов объединены в программно-доступный регистр слова признаков.

Модули исполнительного устройства (микропрограммный, обработки порядка, арифметический, программного сдвига) и регистры временного хранения программно-недоступны для программиста и пользователя, поэтому подробное описание их работы в данной статье не приводится.

Микропрограммный модуль управляет и синхронизирует все модули

и регистры при выполнении команд арифметических и трансцендентных, пересылки данных, сравнения и проверки, а также загрузки констант и других операций.

Модули обработки порядка и арифметический производят все операции над порядком и мантииссой. Арифметический модуль включает в себя 69-разрядный сумматор, регистры, схему округления и другие узлы.

Модуль программного сдвига выполняет левые и правые сдвиги в диапазоне 0...63 двоичных разрядов за один период синхронизации.

Временные регистры — это три 80-разрядных двоичных регистра, каждый из которых состоит из трех значащих полей: знака числа, порядка и

Диапазон изменения и точность представляемых чисел для трех классов данных

Класс данных	Формат	Представляемое десятичное число	
		Диапазон изменения (приближенно)	Точность (число десятичных цифр)
Целочисленные	СЦ (2-байтный)	$-2^{15} \leq X \leq +2^{15} - 1$	4...5
	КЦ (4-байтный)	$-2^{31} \leq X \leq +2^{31} - 1$	9
	ДЦ (8-байтный)	$-2^{63} \leq X \leq +2^{63} - 1$	18
Упакованные двоично-десятичные	УД (10-байтный)	$-10^{18} - 1 \leq X \leq +10^{18} - 1$	18
Действительные	КД (4-байтный)	$2^{-126} \leq X \leq 2^{128}$	6...7
	ДД (8-байтный)	$2^{-1022} \leq X \leq 2^{1024}$	15...16
	ВД (10-байтный)	$2^{-16382} \leq X \leq 2^{16384}$	19...20

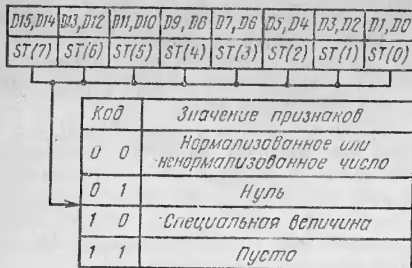


Рис. 11. Формат регистра признаков

мантиссы; длина поля знака — один двоичный разряд, поля порядка — 15 двоичных разрядов и поля мантиссы — 64 двоичных разряда. Эти регистры предназначены для временного хранения данных при различных вычислениях и преобразованиях форматов из внутреннего 10-байтного действительного формата во внешние 2-, 4-

и 8-байтные целочисленные форматы, 10-байтный упакованный двоично-десятичный в 4-, 8- и 10-байтный действительные форматы, а также при преобразовании внешних форматов во внутренний формат.

Представление данных сопроцессора

В сопроцессоре KM1810BM87 используются три класса данных: целочисленные, упакованные двоично-десятичные и действительные (рис. 12, табл. 3). Основной класс данных — действительные. Над целочисленными данными выполняется ограниченное число операций; над упакованными двоично-десятичными данными — только две операции (загрузка и запоминание).

Целочисленные данные — положительные и отрицательные целые числа, нуль, неопределенность. Формат их состоит из двух полей (табл. 4 и 5). В случае целых чисел значение «0» в поле «Знак» соответствует положительному числу, значение «1» — отрицательному. Поле «Модуль» содержит дополнительный код модуля числа. Максимальное отрицательное целое число используется в качестве целочисленной неопределенности при маскированном ответе на особый случай «Недействительная операция».

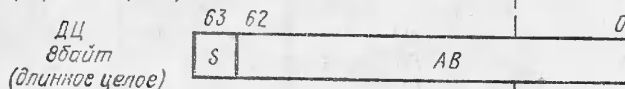
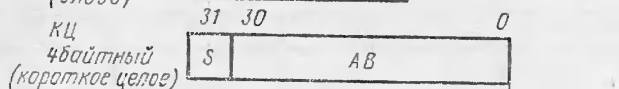
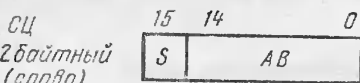
Упакованные двоично-десятичные данные (УД) имеют длину 80 разрядов (10 байтов). В указанный класс данных

Таблица 4

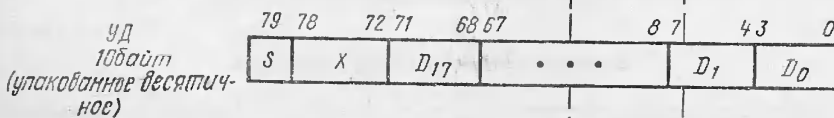
Форматы целочисленных данных

Название и обозначение поля	Размер поля, двоичн. разр.		
	СЦ	КЦ	ДЦ
Знак S	1	1	1
Модуль АВ	15	31	63

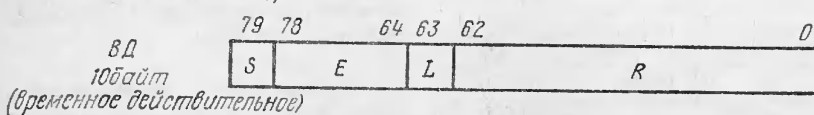
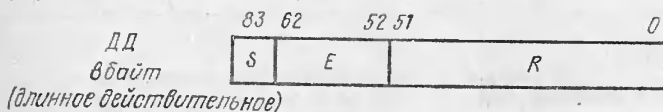
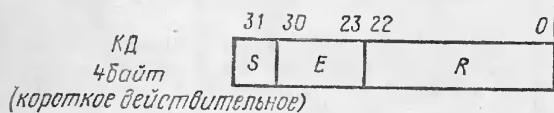
1. Целочисленные данные



2. Упакованные двоично-десятичные данные



3. Действительные данные



S - поле знака; АВ - поле модуля; X - свободное поле; D_i - разряд двоично-десятичной цифры в поле модуля; E - поле порядка; R, L - поле дробной и целой частей мантиссы соответственно.

Рис. 12. Форматы данных

Таблица 5
Представление целочисленных данных в двоичном коде

Вид данного	Содержимое полей в формате данного		
	Знак	Модуль	
		минимум	максимум
Положительное целое число Нуль	0	0...01	1...1
Отрицательное число	1	1...1	0...0
Неопределенность (частный случай отрицательного числа)			0...0

входят положительные и отрицательные целые числа, нуль (со знаком), неопределенность. Формат их состоит из трех полей (см. рис. 12, табл. 6 и 7).

В случае целых чисел значение «0» в поле «Знак» соответствует положительному числу, значение «1» — отрицательному. Полю «Свободно» при запоминании присваивается значение «0», при загрузке это поле игнорируется. Поле «Модуль» содержит модуль числа, состоящий из 18 двоично-десятичных цифр.

Таблица 6
Обозначения и размеры полей УД

Название и обозначение поля	Размеры поля, двоичн. разр.
Знак S	1
Свободно X	7
Модуль D17...D0	4 × 18 = 72

Упакованная двоично-десятичная неопределенность — результат маскированного ответа на особый случай *Недействительной операции* при выполнении команды «Запоминание УД и выгрузка содержимого текущего регистра из стека» (FBSTR). При загрузке неопределенности (см. табл. 6) результат выполнения команды «Загрузка в стек УД» (FBLD) не определен.

Таблица 7
Представление упакованных двоично-десятичных данных в двоичном коде

Вид данного	Содержимое полей в формате данного		
	Знак	Свободно	Модуль
			минимум
Нуль	0/1	0...0	0000, ..., 0000
Целое число	0/1	0...0	0000, ..., 0000, 0001 1001, ..., 1001
Неопределенность	1	1...1	1111, 1111, XXXX, ..., XXXX

Символ X — произвольное значение разряда в поле «Модуль».

не. Ненормализованные числа и псевдонули существуют только в 10-байтном действительном формате. Значения тегов вещественных данных в регистре стека приведены ранее.

Поле «Мантисса» содержит двоичный код значения данного, состоящий из целой и дробной частей. Значение целой части 4- и 8-байтных данных устанавливается неявно. Положение запятой дроби представляемого числа определяется порядком числа.

Установка МП KM1810VM87 в исходное состояние

В исходное состояние МП KM1810VM87 устанавливается аппаратным и программными способами. Аппаратный способ установки МП осуществляется с помощью входного сигнала SR (табл. 11). На рис. 13 показана временная диаграмма синхронизации входного сигнала SR относительно сигнала S и включения источника питания. В процессе работы (перезапуск МП) в исходное состояние KM1810VM87 устанавливается путем подачи на вход SR положительного импульса длительностью не менее четырех периодов сигнала синхронизации S. При поступлении сигнала SR МП пре-

Таблица 8
Размеры полей действительных данных

Название и обозначение поля	КД	ДД	ВД
Знак S	1	1	1
Порядок E	8	11	15
Мантисса, часть:			
целая L	(1)	(1)	1
дробная M	23	52	63

Примечания. 1. В скобках записан неявный разряд целой части мантиссы. 2. Поле «Знак» определяет знак данного: содержимое «0» — данное положительное, содержимое «1» — данное отрицательное. 3. Поле «Порядок» содержит смещенный порядок данного $E = P - B$ (см. табл. 9).

Таблица 9
Значения данных для поля «Порядок»

Название и обозначение	КД	ДД	ВД
Несмещенный порядок	R_{\max} 128 R_{\min} —127	1024 —1023	16384 —16383
Смещение	B 127 (7FH)	1023 (3FFFH)	16383 (3FFFH)
Смещенный порядок	E_{\max} 255 (FFH) E_{\min} 0 (00H)	2047 (7FFFH) 0 (000H)	32767 (7FFFH) 0 (0000H)

крашает выполнение внутренних операций и с задержкой на один период сигналов синхронизации S переводит выводы в состояние:

Вывод	Состояние вывода
A/D15 ... A/D0	Высокоимпедансное
A19/SA6 ... A16/SA3	»
BHE/SA7, SA2...SA0	Переход через высокий уровень в высокоимпедансное
$\overline{RQ}/GT0$	«1»
RQ/GT1	«1»
INT	«0»
BUSY	«0»

Таблица 10
Представление действительных данных в двоичном коде

Вид данного	Формат данного	Содержимое полей в формате данного						
		Знак	Порядок		Мантисса			
			мин	макс	мин	макс		
Нуль (истинный)	4, 8, 10	0/1	0...0		[0], 0...0			
Денормализованное число	4, 8, 10	0/1	0...01		[0], 0...01	0, 1...1		
	10				1, 0...01	1, 1...1		
Псевдонуль (как ненормализованное число)	10	0/1	0...01		0, 0...0			
Ненормализованное число (снизу)	10				1...10	0, 0...01	0, 1...1	
Нормализованное число	4, 8, 10					[1], 0...0	1, 1...1	
Псевдонуль (как нечисло)	10					0, 0...0		
Ненормализованное число (сверху)	10					0, 0...01	0, 1...1	
Бесконечность	4, 8, 10				1...1	1, 0...0		
Нечисловая величина	4, 8, 10						[1], 0...01	1, 1...1
Неопределенность (частный случай нечисла)	4, 8, 10				1	1...1	[1], 10...0	

Примечание. В квадратных скобках указан необязательный разряд целой части мантисы данного. Для 10-байтных данных этот разряд указывается явно, для 4- и 8-байтных — его наличие предполагается.

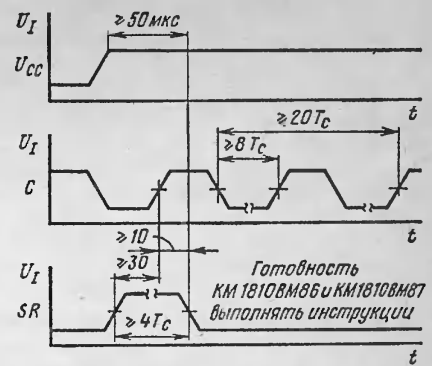


Рис. 13. Синхронизация включения и сброса КМ1810ВМ87

первую выборку осуществляет ЦП КМ1810ВМ88, то на входе ВНЕ/SA7 устанавливается напряжение высокого уровня, так как выбирается байт команды, расположенной по четному адресу ВНЕ=1 при АО=0.

Через $20 \cdot T_c$ после перехода сигнала SR в состояние низкого уровня МП КМ1810ВМ87 готов к выполнению команд. В исходное состояние МП КМ1810ВМ87 устанавливается программным способом с помощью команд FINIT/FNINIT и FSAVE, но в отличие от аппаратного способа без опознавания типа ЦП.

Параметры и режимы эксплуатации

Статические (табл. 12) и динамические (рис. 14 и 15) параметры МП КМ1810ВМ87 приведены при $U_{cc} = +5 \text{ В} \pm 5\%$ и температуре окружающей среды от -10 до 70° . В активном режиме все сигналы, которые управляют местной шиной, вырабатывает МП КМ1810ВМ87, а в пассивном режиме — ЦП КМ1810ВМ86 или КМ1810ВМ88.

Таблица 11

Состояние внутренних регистров КМ1810ВМ87 после аппаратного способа установки в исходное состояние

Регистр	Значение в двоичной системе исчисления	Интерпретация
Управляющее слово:		
режим бесконечности	0	Проективный
режим округления	00	Округление к ближайшему числу
режим точности	11	64 разряда
маска разрешения прерывания	1	Прерывания запрещены
маски особых случаев (ОС)	111111	Все ОС маскированы
Слово состояния:		
занято	0	Не занято
код условия	Не определен	Не определен
верхняя позиция стека	000	Стек очищен
запрос прерывания	0	Прерывания отсутствуют
флажки особых случаев	000 000	Особые случаи отсутствуют

При аппаратной установке исходного состояния МП КМ1810ВМ87 автоматически определяет тип ЦП с помощью двунаправленного вывода ВНЕ/SA7, который подсоединяется к одноименному выводу микросхемы КМ1810ВМ86 или к выводу SS0 микросхемы КМ1810ВМ88.

По истечении восьми периодов сигнала синхронизации C ($8 \cdot T_c$) после перехода сигнала SR в состояние низкого уровня (см. рис. 13) ЦП выбирает слово первой команды из ячейки памяти по адресу FFFF0H. Если микросхема КМ1810ВМ86 является ЦП, то на входе ВНЕ/SA7 во время выдачи адреса FFFF0H устанавливается напряжение низкого уровня, так как первая выборка соответствует слову с четным адресом ВНЕ=0 при АО=0. Если в первом цикле шины

Регистр	Значение в двоичной системе исчисления	Интерпретация
Слово признаков:	11	Пусто
признаки	Без изменений	Без изменений
Регистровый стек		
Указатели текущей команды:		
код команды	—«—	—«—
адрес команды	—«—	—«—
адрес операнда	—«—	—«—

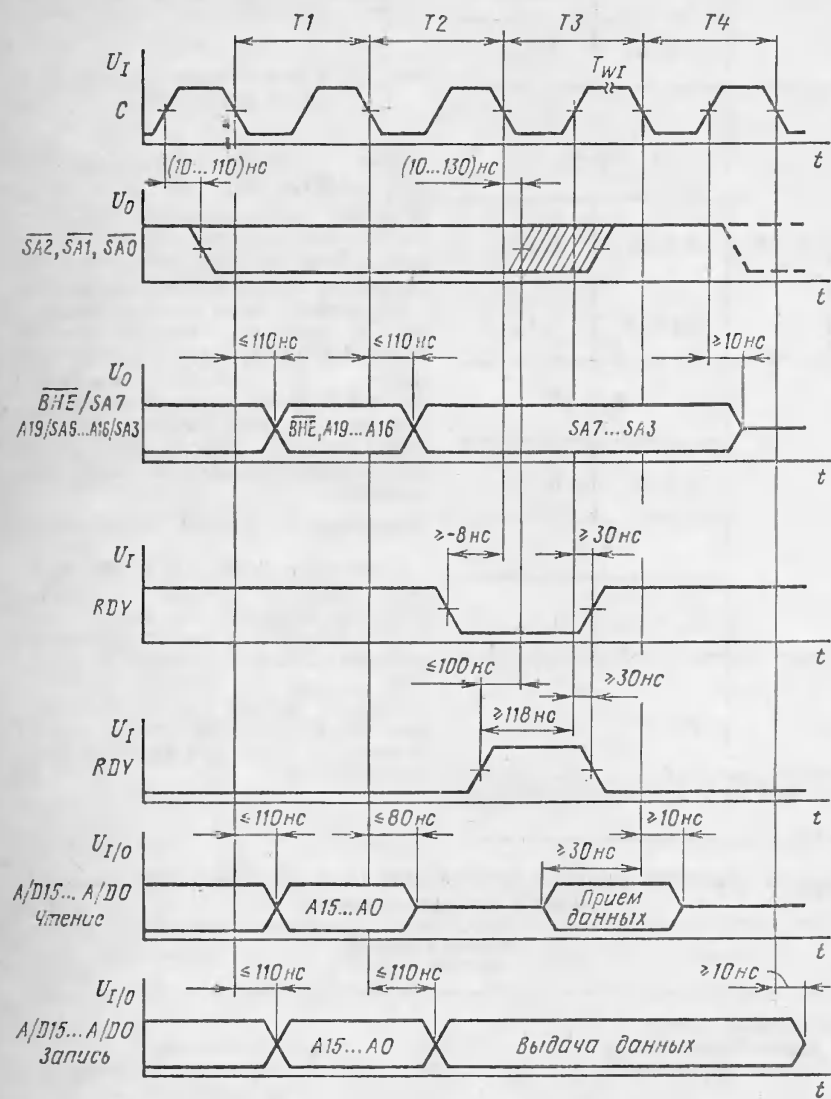


Рис. 14. Синхронизация управляющих и выходных сигналов в активном режиме работы МП КМ1810ВМ87

Статические параметры МП КМ1810ВМ87 Таблица 12

Параметр	Обозначение	Норма	
		не менее	не более
Напряжение питания, В	U_{CC}	4,75	5,25
Входное напряжение, В высокого уровня	U_{IH}	2,0	$U_{CC} \pm 0,5$
низкого уровня	U_{IL}	-0,5	

Процесс организации доступа МП КМ1810ВМ87 к использованию местной шины, когда ею управляет ЦП КМ1810ВМ86 или КМ1810ВМ88, изображен на рис. 16, а когда ею управляет МП КМ1810ВМ87, — на рис. 17. На рис. 18 показана синхронизация включения сигналов INT и BUSY относительно сигнала С.

Включение МП КМ1810ВМ87 в вычислительную систему (рис. 19)

Входные сигналы Q51 и Q50 МП КМ1810ВМ87, поступающие с одноконтурных выходов ЦП КМ1810ВМ86 или КМ1810ВМ88, позволяют выбирать и декодировать команды синхронно с ЦП. Из общего потока команд сопроцессор выбирает только команду расширения ESC ЦП. Адресная часть команды ESC обрабатывается только ЦП, команде ESC должна предшествовать команда WAIT для предотвращения выборки команд ESC до окончания обработки сопроцессором предыдущей команды. Если необходимость в команде WAIT отсутствует, то для сохранения метрических характеристик объектного кода вместо нее используется команда NOP.

ЦП вычисляет исполнительный адрес операнда в памяти, если в команде определена ссылка на него, затем выполняет фиктивный цикл чтения слова памяти по этому адресу. Прочитанные данные ЦП игнорируются. Сопроцессор запоминает вычисленный ЦП исполнительный адрес. При чтении операнда из памяти устройство управления выбирает первое слово операнда в момент фиктивного цикла чтения, затем самостоятельно считывает последующие слова (если необходимо).

При записи результата в память устройство управления не учитывает данные, полученные в момент фиктивного цикла чтения. Если сопроцессор сформировал результат выполнения операции, который должен быть записан в память, то устройство управления записывает его в память по предварительно сохраненному адресу. Число обращений к памяти зависит от числа слов в результате, а при фиктивном цикле чтения операнда — от типа ЦП, который определяется автоматически по сигналу системного сброса: для 16-разрядного — одно обращение; для 8-разрядного — два обращения.

Если сопроцессор обращается к операнду в памяти по нечетному адресу, то 16-разрядный ЦП оптимизирует обмен с памятью посредством обращения к одному (старшему) байту слова памяти и последующего обращения к двум байтам слова памяти. Все операнды команд сопроцессора в памяти занимают четное число байтов. Для увеличения производительности вычислительной системы на базе 16-разряд-

ного ЦП рекомендуется размещать данные сопроцессора по четным адресам. Пример выполнения последовательности команд ЦП и сопроцессором показан на рис. 20.

Выходной сигнал BUSY МП КМ1810ВМ87 информирует ЦП о том, что в данный момент занят (не занят) выполнением программы. Сигнал BUSY поступает на вход TEST ЦП, который с помощью команды WAIT проверяет этот сигнал. Если на входе TEST присутствует напряжение низкого уровня (МП КМ1810ВМ87 не выполняет команд), то ЦП продолжает выборку команд из программы, в противном случае он переходит в состояние ожидания, так как МП КМ1810ВМ87 занят обработкой команды, результат которой необходим для использования в следующей команде.

МП КМ1810ВМ87 может осуществлять прерывания ЦП с помощью выходного сигнала INT при обнаружении им немаскированного особого случая. Вывод INT (см. рис. 19) подключается к ЦП через программируемый контроллер прерываний КР1810ВН59А.

Для доступа к управлению местной шиной при приеме или передаче данных в МП КМ1810ВМ87 двунаправленный вывод RQ/GT0 (запрос/разрешение) подключается к одному из выводов RQ/GT0 или RQ/GT1 ЦП. Второй вывод RQ/GT1 МП КМ1810ВМ87 используется в микропроцессорной системе для получения доступа к управлению местной шиной МП ввода-вывода КМ1810ВМ89. Этот вывод имеет более высокий приоритет по сравнению с выводом RQ/GT0.

Сигналы SR, C и RDY (сброс, синхронизация и готовность соответственно) синхронизируются в генераторе фаз КР1810ГФ84 и поступают на одноименные входы ЦП КМ1810ВМ86 или КМ1810ВМ88, МП КМ1810ВМ87 и МП ввода-вывода КМ1810ВМ89 для установки их в исходное состояние, синхронизации с контроллером системной шины КР1810ВГ88, а также для подтверждения адресуемым устройством окончания (неокончания) обмена с памятью в текущем цикле местной шины соответственно.

Двунаправленный вывод BHE/SA7 (разрешение передачи данных по старшей половине местной шины/состояние) выполняет несколько функций: входной сигнал BHE определяет тип ЦП в T1 первого цикла местной шины после выполнения аппаратного сброса; в последующих циклах местной шины сигнал BHE является выходным и разрешает передачу данных по старшей

Параметр	Обозначение	Норма	
		не менее	не более
Входное напряжение тактовых импульсов, В			
высокого уровня	U_{CIN}	3,9	$U_{CC}+1,0$
низкого уровня	U_{CIL}	-0,5	0,6
Выходное напряжение, В			
высокого уровня	U_{OH}	2,4	—
низкого уровня	U_{OL}	—	0,45
Ток			
потребления, мА	I_{CC}	—	540
утечки на входах, мкА	I_{IL}	—	$ \pm 10 $
выходной в состоянии «выключено», мкА	I_{OZ}	—	$ \pm 10 $
Емкость, пФ			
входов и выходов BUSY, INT	C_1	—	10
входов-выходов и тактового сигнала C	$C_{I/O}$	—	15
нагрузки на выводах RQ/GT0, RQ/GT1	$C_{L(RQ/GT)}$	—	40
нагрузки на остальных выводах	C_L	—	100

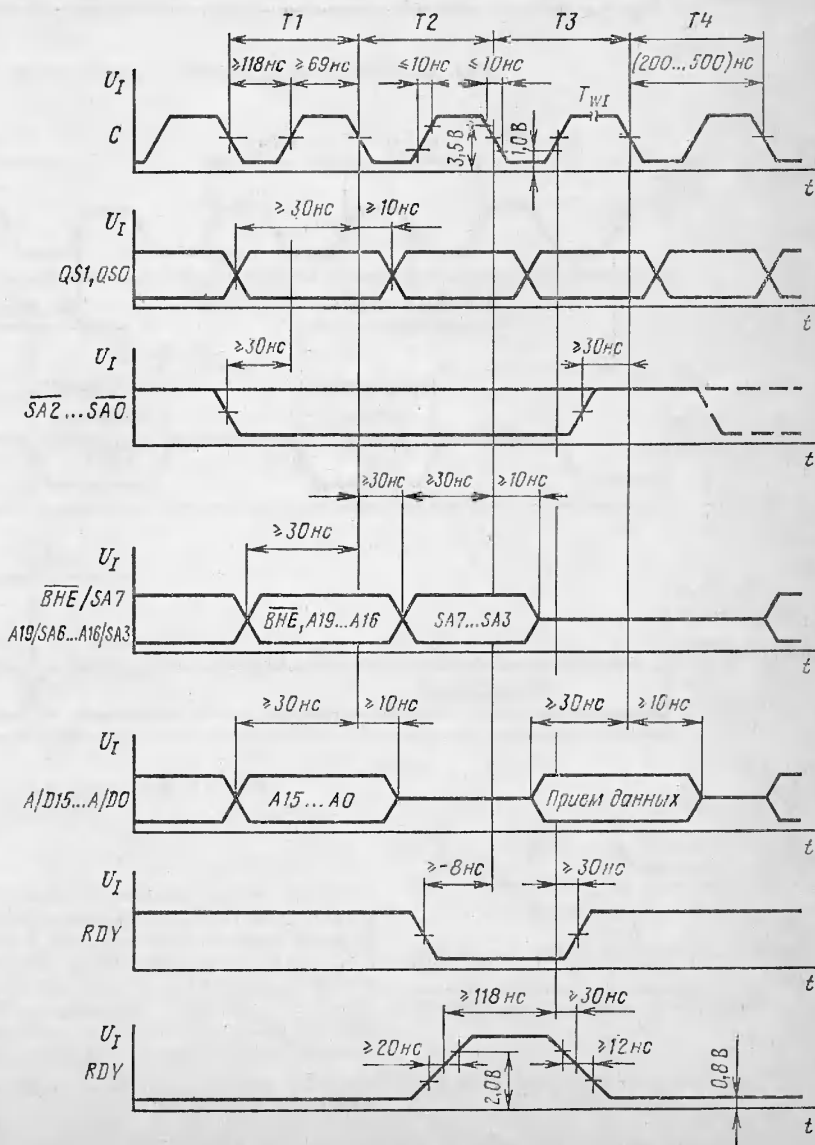


Рис. 15. Синхронизация управляющих и выходных сигналов в пассивном режиме работы МП КМ1810ВМ87

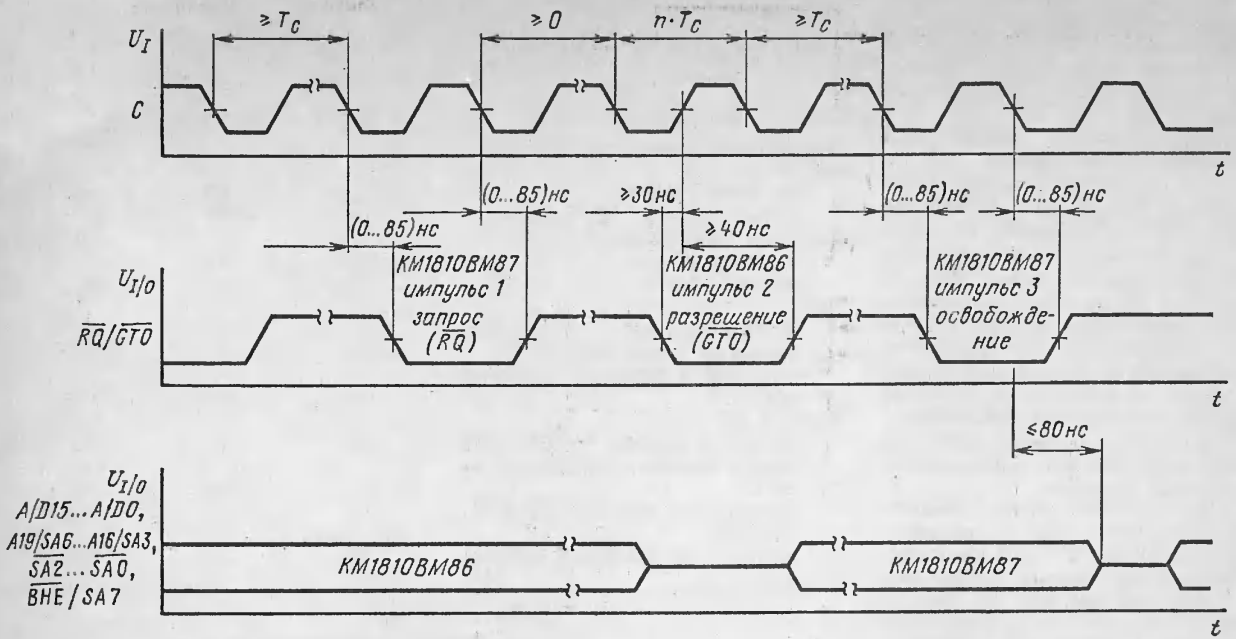


Рис. 16. Временные диаграммы синхронизации сигналов $\overline{RQ}/\overline{GT0}$

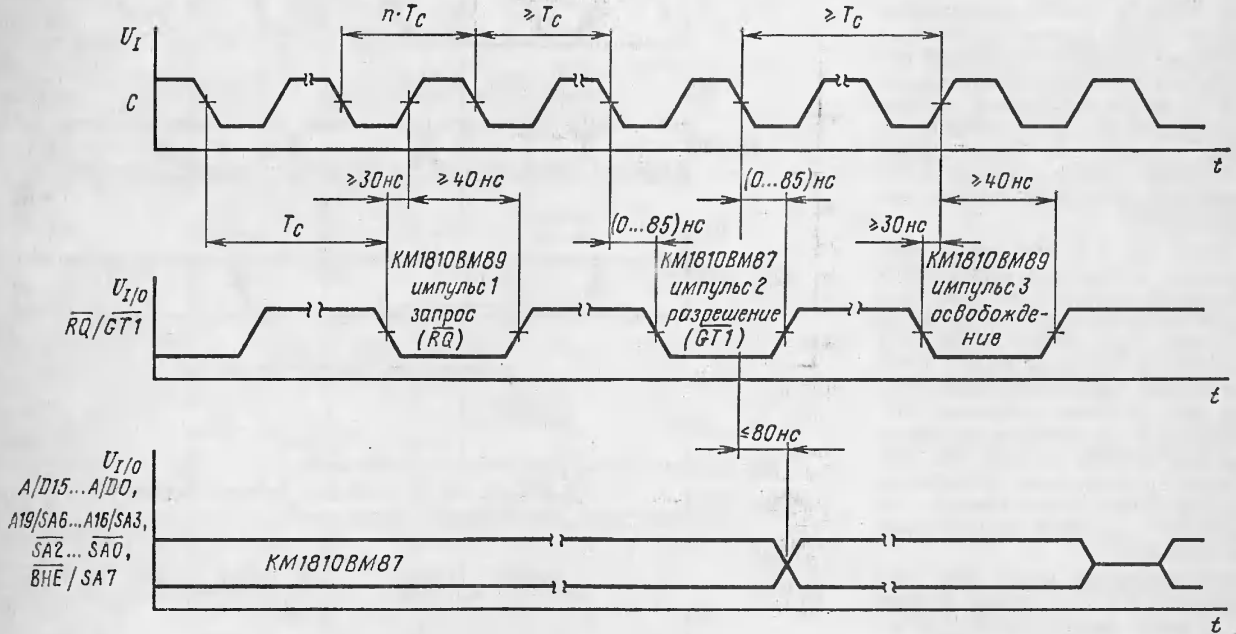


Рис. 17. Временные диаграммы синхронизации сигналов $\overline{RQ}/\overline{GT1}$

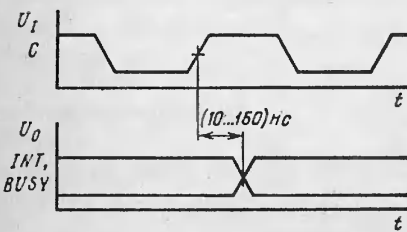


Рис. 18. Синхронизация сигналов INT и BUSY

половине местной шины. Сигнал SA7 в тактах T2, T3, T_{W1} и T4 выдает (принимает) информацию состояния. Вывод SA7 является входом, когда местной шиной управляет ЦП, и выходом, когда местной шиной управляет МП KM1810BM87. По уровню входного (выходного) напряжения сигнал SA7 соответствует сигналу ВНЕ того же цикла местной шины.

После предоставления ЦП местной шины в распоряжение МП KM1810BM87 последний вырабатывает сигналы состояния SA2... SA0. Программно управляемые двунаправленные выходы SA2... SA0 МП KM1810BM87 подключаются к контроллеру KP1810BG88, вырабатывающему управляющие сигналы для организации доступа к памяти и устройствам ввода-вывода.

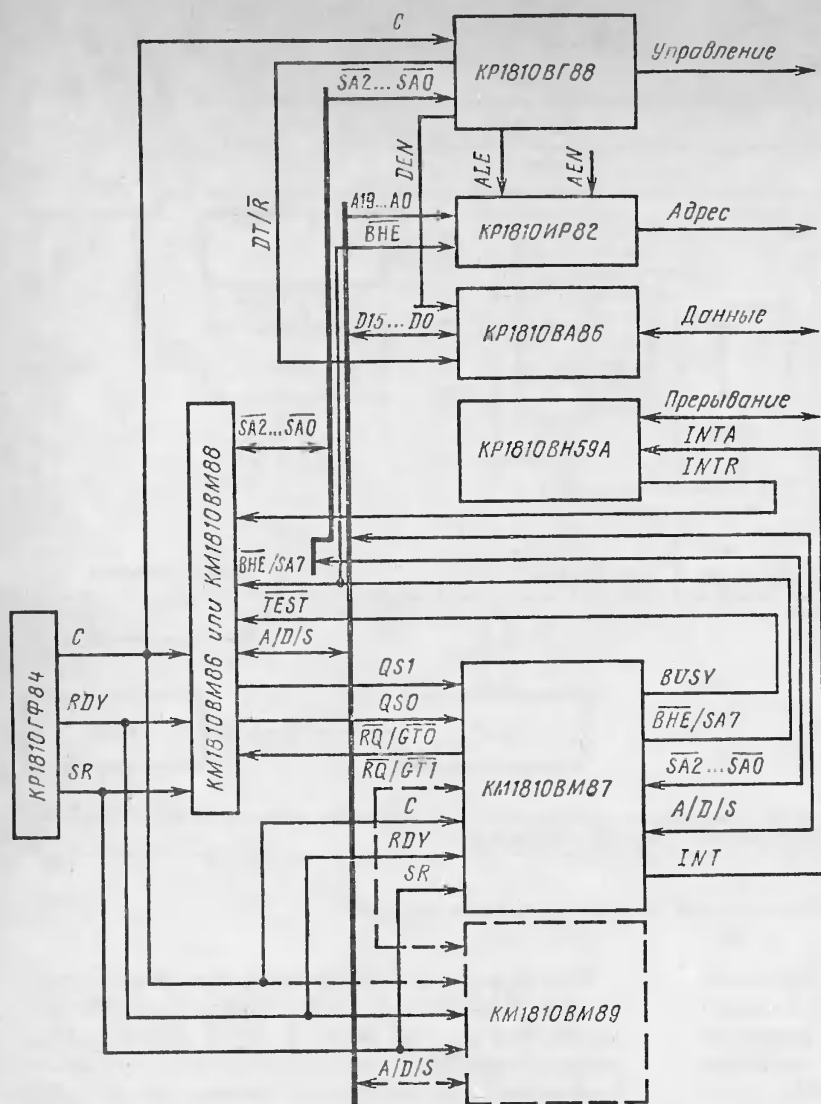


Рис. 19. Включение МП KM1810BM87 в вычислительную систему

Выводы A/D/S двунаправленные трехстабильные ЦП KM1810BM86 или KM1810BM87 и МП ввода-вывода KM1810BM89 образуют местную шину, к которой при необходимости каждый из них имеет доступ. Когда местной шиной управляет, например, ЦП, то выводы A/D/S двух других МП нахо-

дятся в высокоимпедансном состоянии. Выдача (прием) информации адреса, данных или состояния по выводам A/D/S разнесена во времени, и при записи в шинные формирователи KM1810BA86 (KM1810BA87) или регистры адреса KM1810IP82 (KM1810IP83) происходит ее стро-

Команды программы	Выполнение команд ЦП	Сопроцессор	Сигнал
FMUL	WAIT ESC	TEST-BUSY	
FSORT	WAIT ESC	TEST-BUSY	FMUL
CMP ALPHA, 100	CMP		FSQRT
JG CONT	JG		
MOV ALPHA, 100	MOV		
CONT: FIST BETA	WAIT ESC	TEST-BUSY	
FWAIT	WAIT	TEST-BUSY	FIST
MOV AX, BETA	MOV		

Рис. 20. Пример выполнения последовательности команд ЦП и сопроцессором

бирование соответствующими управляющими сигналами, которые вырабатывает контроллер системной шины KM1810BF88. Сигналы A19...A0 и BHE записываются в регистры адреса и выдаются на шину адреса сигналом ALE, а обмен данными между местной шиной и системной шиной осуществляется с помощью управляющих сигналов DT/R и DEN контроллера KM1810BF88.

Шины данных, адреса, управления и прерывания образуют системную шину, а показанная на рис. 19 схема — микропроцессорный модуль, который представляет собой мощную вычислительную систему. Данные по системной шине передаются одинаково у МП KM1810BM87 и ЦП; сигналы состояния, адреса и данных полностью идентичны.

Телефон 442-95-03, 434-95-11, Киев

Статья поступила 2.12.88

УДК 681.322—192

Б. В. Антонок, И. В. Омельчук, В. Н. Присяжнюк, Г. Г. Терещенко

ТЕСТОВЫЙ ПРОЦЕССОР ДЛЯ МУЛЬТИПРОЦЕССОРНЫХ СИСТЕМ

Одна из основных задач проектирования высоконадежных МПС — повышение уровня контролепригодности процессоров — может быть решена введением в состав процессора аппа-

ратно-программных средств контроля [1—3].

Результаты вычислений могут сравниваться как с помощью специального диагностического процессора [2, 4], так и самими процессорами диагностируемой структуры [5, 6]; в мультипроцессорных системах с распределенной магистралью — в процессе передачи по магистрали.

На базе МПК БИС серии K1810 разработан процессор, обеспечивающий селективный конт-

ший байт слова используется для хранения эталонной информации, подлежащей передаче по СМ, старший байт — для записи информации, переданной по СМ в контролируемый сектор.

КИСМ предназначен для селективного контроля информации, передаваемой по СМ. Селективность контроля обеспечивается делением пространства, адресуемого по СМ, на группу секторов. Старшие разряды СМ адреса интерпретируются при этом как номер сектора, а младшие — как смещение в секторе. КИСМ контролирует передачу информации по адресам, старшие разряды которых совпадают с выбранным номером сектора. Номер контролируемого сектора устанавливается программным способом. КИСМ содержит программируемый узел управления (ПУ), узел захвата-освобождения РМ (ЗОМ) и узел записи и чтения со сравнением (ЗЧС).

Узел ПУ включает программно-доступные регистры номера контролируемого сектора (НКС) и фиксации магистрали арбитража (ФМА), схему разрешения-запрета контроля (РЗК) и схему управления (СУ1).

Разрешение и запрет работы ПУ и, следовательно, блока КИСМ обеспечиваются программным способом: ПУ сравнивает номера секторов текущих передач с выбранным номером сектора; при передачах в контролируемый сектор фиксирует состояние магистрали арбитража (АРБ) в регистре ФМА и вырабатывает сигналы управления узлами ЗОМ и ЗЧС.

Узел ЗОМ передает блоку МП сигнал запроса РМ, принимает от него сигнал разрешения РМ и вырабатывает сигнал управления узлом ЗЧС. После завершения использования РМ узел ЗОМ передает блоку МП сигнал разрешения магистрали.

Узел ЗЧС содержит регистр адреса (РА), регистр данных (РД), программно-доступный регистр базы (РБ), схему сравнения (СС) и схему управления (СУ2). РБ используется для хранения начального адреса обменного сектора. Его состояние изменяется программным способом, что обеспечивает возможность произвольного размещения обменного сектора в блоке ЛП. Кроме того, узел ЗЧС фиксирует по сигналу управления от узла ПУ состояние магистрали данных и младших разрядов магистрали адреса в РД и РА и блокирует работу узла ПУ. По сигналу управления от узла ЗОМ узел ЗЧС устанавливает на РМ адрес, старшие разряды которого определяются состоянием РБ, а младшие — РА, и вырабатывает для схемы управления блока ЛП сигналы чтения младшего банка и записи в старший банк.

При несовпадении информации, считанной из младшего банка, с информацией, переданной по СМ и хранимой в РД, узел ЗЧС вырабатывает сигнал немаскируемого прерывания и за-

прещает узлу ПУ изменять состояние регистра ФМА. После завершения шинного цикла записи и чтения узел ЗЧС снимает сигнал блокирования узла ПУ.

Рассмотрим работу процессора в составе мультипроцессорной системы с распределенной магистралью. В исходном состоянии работа КИСМ запрещена, а блок МП выполняет задания, не связанные с контролем информации, передаваемой по распределенной магистрали. Информационным признаком начала режима контроля является сигнал запроса прерывания (ЗПР), поступающий на блок ПКП. Источником ЗПР при периодическом контроле является блок ПТ или внешние устройства. По сигналу прерывания (ПРЕ) блок МП переходит к выполнению подпрограммы обслуживания прерывания. При этом он определяет номер контролируемого сектора и выделяет в ЛП обменный сектор, в который заносятся результаты вычислений. Размерность обменного сектора равна удвоенной размерности контролируемых по системной магистрали секторов.

Каждому байту контролируемого адресного пространства соответствует в обменном секторе слово. В младший банк ОЗУ1 блока ЛП заносится байт прямого значения информации, а в старший банк ОЗУ2 — инверсного. Смещение записываемого слова относительно начального адреса обменного сектора равно смещению контролируемого байта относительно начала сектора в пространстве адресов СМ. Блок МП через РМ управления и данных заносит в регистр РБ узла ЗЧС базовый адрес обменного сектора ЛП, а в регистр НКС узла ПУ (рис. 2) — номер контролируемого сектора, разрешает работу блока КИСМ, а затем возобновляет прерванную программу.

В шинных циклах обращений в область адресов контролируемого сектора КИСМ контролирует информацию, передаваемую по распределенной СМ. При этом СУ1 узла ПУ вырабатывает сигнал СТБ высокого уровня, поступающий в узлы ЗОМ и ЗЧС. Сигналом СТБ в регистре ФМА узла ПУ фиксируется состояние СМ арбитража, а в регистрах РА и РД узла ЗЧС — состояние СМ адреса и младшей половины СМ данных соответственно. По сигналу СТБ узел ЗОМ начинает операцию захвата РМ. На входе-выходе ЗПР/РАЗР формируется импульс 1 (рис. 3), сигнализирующий о том, что МП должен отключиться от РМ. Блок МП, завершив текущий шинный цикл, освобождает РМ, сигнализируя об этом импульсом 2 по входу-выходу ЗПР/РАЗР.

Импульсами ЗПР/РАЗР в узлах ПУ и ЗЧС блокируется обновление регистров РА, РД и ФМА, после чего узел ЗОМ формирует сигнал ВКЛ низкого уровня, разрешающий подключение узла ЗЧС к РМ. Узел ЗЧС устанавливает на магистрали РМА адрес слова обменного сек-

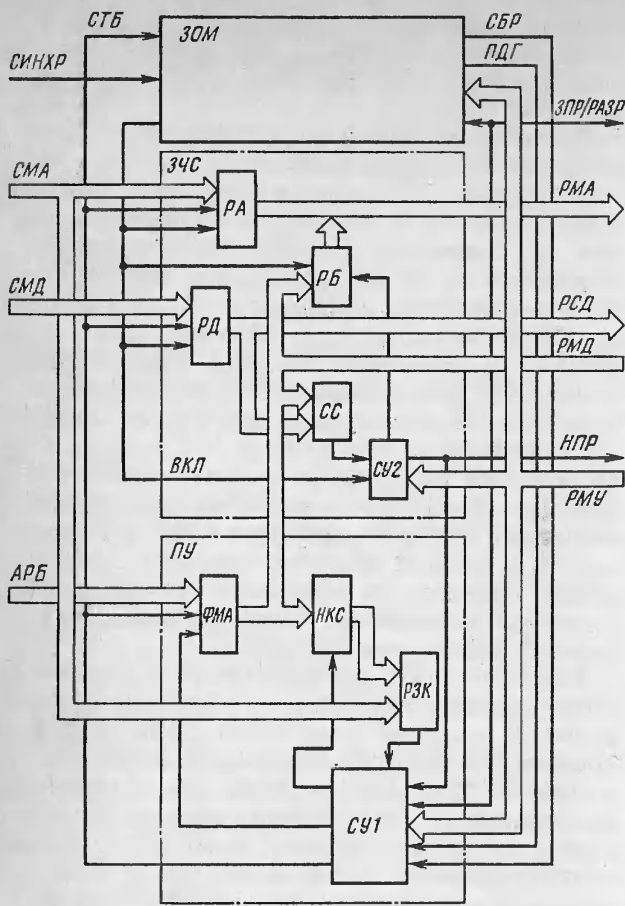


Рис. 2. Схема контроля данных, передаваемых по систем-ной магистрали

тора, а на магистрали РМУ — сигналы чтения и записи.

Схема управления СУ блока ЛП по линиям управления (ЛУ) записывает в банк ОЗУ2 информацию, поступающую по старшей половине РМ данных РСД из регистра РД, и считывает эталонную информацию из банка ОЗУ1. Таким образом, инверсный байт информации замещается байтом информации, переданной по распределенной магистрали. Считанная из банка ОЗУ1 эталонная информация по младшей половине РМ данных поступает в узел ЗЧС, где срав-

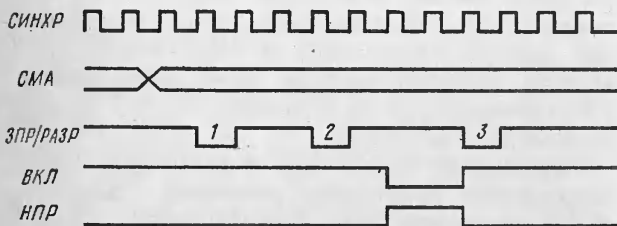


Рис. 3. Временная диаграмма работы блока КИСМ

«Микропроцессорные средства и системы» № 5, 1989

нивается с информацией, хранимой в регистре РД.

После завершения операций записи и чтения со сравнением узел ЗОМ снимает активный низкий уровень сигнала ВКЛ, отключая узел ЗЧС от РМ, и импульсом 3 (рис. 3) на входе-выходе ЗПР/РАЗР сообщает об этом блоку МП. В следующем такте узел ЗОМ формирует на выходе СБР сигнал снятия блокировки регистров РА и РД узла ЗЧС и регистра ФМА узла ПУ. Если передаваемая информация идентична эталонной, то операция контроля информации «прозрачна» для МП.

В случае несовпадения информации, переданной по распределенной магистрали, с информацией, хранимой в ОЗУ1, на выходе схемы сравнения (СС) появляется сигнал высокого уровня, приводящий к формированию на выходе узла ЗЧС сигнала немаскируемого прерывания (НПР) высокого уровня, поступающего на вход блока МП и на узел ПУ, запрещающий обновление регистра ФМА.

Неравенство содержимого старшего и младшего банков блока ЛП считается информационным признаком того, что по распределенной магистрали передана недостоверная информация. В подпрограмме обслуживания немаскируемого прерывания посредством считывания содержимого регистра ФМА узла ПУ определяется номер процессора, выполнившего первую недостоверную передачу, и вырабатывается команда на его блокировку. После завершения подпрограммы обслуживания НПР снимается запрет на обновление ФМА. Анализ состояния обменного сектора позволяет определить и в некоторых случаях устранить последствия отказа.

Таким образом, функциональные возможности тестового процессора обеспечивают обнаружение неисправных процессов мультипроцессорной системы и фиксацию протокола передач в контролируемое адресное пространство. Режим контроля управляется программно. Анализируя состояние обменного сектора, можно установить выполнение достоверных или недостоверных передач или отсутствие передач.

Использование ОС с объектно-ориентированной архитектурой позволяет реализовать указанные функции посредством управления объектом «контролируемый сектор» (КС). Атрибутами этого объекта служат начальный адрес и размерность самого объекта, начальный адрес и состояние обменного сектора. Системные вызовы управления объектом КС обеспечивают его создание, удаление и проверку состояния. Если используется ОС RMX-86 [9], то КС станет дополнительным объектом ядра ОС, а системные вызовы — расширениями ОС. 252051, ул. Маршала Рыбалко, 8/10, НПО «Электрон-прибор»; тел. 274-42-86

1. Алексенко А. Г., Глазков М. А., Галицын А. А. Высоконадежная ПЭВМ на базе БИС К1810ВМ86 // Микропроцессорные средства и системы.— 1987.— № 1— С. 21—27.
2. Волочий Б. Ю., Калашников И. Д., Мазепа Р. Б., Мандзий Б. А. Проектирование отказоустойчивых микропроцессорных информационно-измерительных систем.— Львов: Виша школа, 1987.
3. Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения.— М.: Радио и связь, 1986.
4. Crouzet Yves, Chayade Jacques. A 6800 coprocessor for error detection in microcomputers; the RAD // Proceeding of the IEEE.— 1986.— Vol. 74, N 5.— P. 723—731.
5. Yang Ted C., Liu Jang. A reliable multimicroprocessor system // IECON 84: Pros. Int. Conf. Ind. Electron.

6. Lombardi Fabrizio, Ratheal Steve. Diagnosis of microcomputer system by triplet assertion // Software and Microsystems.— 1984.— Vol. 3, N 5—6.
7. Кобылинский А. В., Москалевский А. И., Темченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86 // Микропроцессорные средства и системы.— 1986.— № 1.— С. 28—33.
8. Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г., Шаронин Ю. Г. Учебная микроЭВМ на основе микропроцессора КМ1810ВМ86 // Микропроцессорные средства и системы.— 1986.— № 5.— С. 16—22.
9. Systems data catalog. Intel. January, 1981.

Статья поступила 18.01.88

УДК 681.325.5

Э. П. Розенштейн, А. В. Овчинников

СИНХРОНИЗАЦИЯ РАБОТЫ МИКРОПРОЦЕССОРОВ СЕРИИ К1801

Микропроцессоры серии К1801 (К1806) имеют нижний предел частоты синхронизации, равный 0 Гц [1—3]. Это означает, что они могут быть остановлены на неопределенное время при снятии частоты. Все внутренние состояния МП сохраняются и, снова подав частоту синхронизации, можно продолжать прерванную работу, что очень удобно при отладке МПУ, ПО, создании стендов для изучения микропроцессорной техники школьниками и студентами.

Простое и удобное устройство для снятия частоты синхронизации изображено на рис. 1. При низком уровне сигнала \overline{STOP} на выходе $CLOCK$ установлен Лог. 0 или Лог. 1 в зависимости от того, в какой фазе частоты синхронизации сигнал \overline{STOP} был переведен в 0 (рис. 2). В начальный момент времени сигнал \overline{STOP} задается высоким уровнем и на тактовый вход МП приходит частота синхронизации (рис. 3). МП выставляет на шину адреса-данных 16-разрядный адрес и стробирует его сигналом \overline{SYNC} . После перехода сигнала \overline{SYNC} из высокого состояния в низкое сигнал \overline{STOP} устанавливается в 0, частота синхронизации снимается и МП останавливается. На шине адреса-данных устанавливается адрес А ячейки памяти, к которой в данном цикле обмена обращается

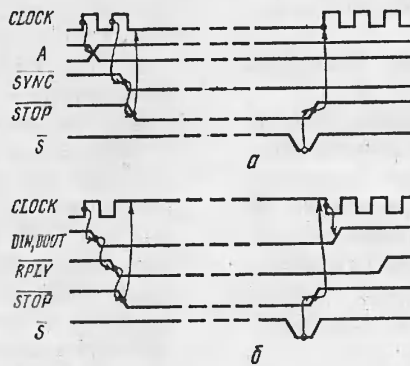


Рис. 2. Временная диаграмма работы устройства для снятия частоты синхронизации МП: частота фиксируется в состояниях Лог. 0 (а) и Лог. 1 (б)

сигнал \overline{SYNC} . При активизации сигнала \overline{RPLY} сигнал \overline{STOP} переводится в Лог. 0, подача частоты синхронизации прекращается, останавливается МП. Теперь на шине адреса-данных будут присутствовать данные. Кроме сигналов \overline{SYNC} и \overline{RPLY} будет активен также один из сигналов \overline{DIN} или \overline{DOUT} .

Телефон 133-80-89, Москва

ЛИТЕРАТУРА

1. Дшхунян В. Л. и др. Однокристалльные микропроцессоры комплекта БИС серии К1801 // Микропроцессорные средства и системы.— 1984.— № 4.— С. 12—18.
2. Волков Р. И. и др. Однокристалльный микропроцессор КМ1801ВМ3 // Микропроцессорные средства и системы.— 1984.— № 4.— С. 12—18.

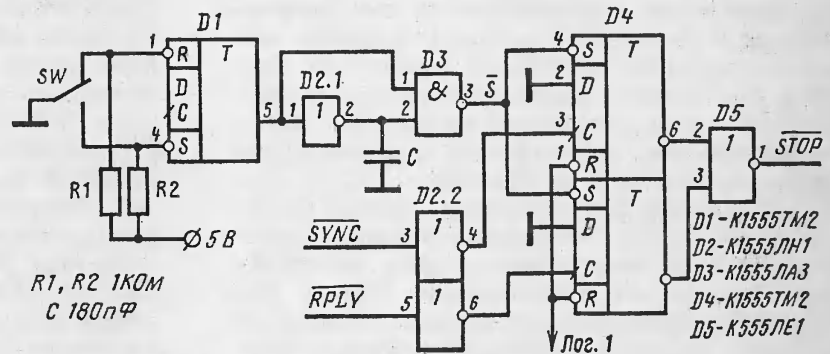


Рис. 3. Схема формирования сигнала \overline{STOP} сигналами \overline{SYNC} и \overline{RPLY}

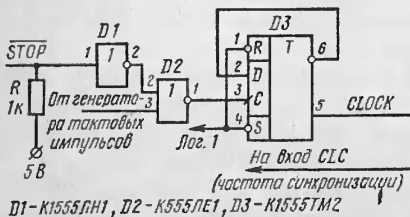


Рис. 1. Устройство для снятия частоты синхронизации МП серии К1801

МП. В таком режиме МП может находиться бесконечно долго. Сигнал \overline{STOP} нажатием на кнопку SW переводится в неактивное состояние, возобновляется подача частоты синхронизации на тактовый вход \overline{CLC} . МП работает до тех пор, пока на него не придет сигнал \overline{RPLY} или пока он не выставит новый

темы.— 1986.— № 4.— С. 37—40.

3. Хвощ С. Т., Варлинский Н. Н., Попов Е. А. Микропроцессоры и микроЭВМ в системах автоматического управления. Справочник.— Ленинград: Машиностроение, 1987.

Сообщение поступило 4.10.88

В. Г. Захаров, Р. Г. Бияшев

ОЦЕНКА ТЕХНИЧЕСКОГО УРОВНЯ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Государственным комитетом СССР по вычислительной технике и информатике утверждена Временная методика оценки технического уровня и качества средств вычислительной техники (СВТ) общего назначения, подготовленная Межведомственной рабочей группой. Методика устанавливает единые правила и порядок оценки технического уровня и качества СВТ в процессе их разработки, серийного производства, при аттестации по двум категориям качества, при проведении ведомственных и межведомственных оценок. Методика обязательна для всех министерств и ведомств — разработчиков и изготовителей СВТ.

Основные положения Временной методики таковы.

1. Технический уровень средств вычислительной техники — это совокупность свойств конкретного изделия, характеризующих функциональное назначение изделия, уровень затрат на его создание и эксплуатацию, а также потребительские, конструкторские, конструктивно-технологические и другие показатели, отражающие техническое совершенство и качество изделия.

2. Технический уровень определяется на основе сопоставительного анализа значений технических характеристик (показателей) оцениваемого образца вычислительной техники, характеризующих указанную выше совокупность свойств в сравнении с соответствующими техническими характеристиками конкретного аналога или базового образца (реально достижимой совокупности значений показателей технического уровня), отражающих мировые достижения и тенденции их развития.

3. Принятые для сравнения аналоги должны иметь то же назначение, условия применения и быть ориентированы на ту же группу потребителей, что и оцениваемое изделие. Недопустимо сравнивать отдельные показатели у разных аналогов. Оценка технического уровня конкретных образцов вновь разрабатываемых СВТ должна опираться на прогнозируемые значения соответствующих показателей аналогов (базовых образцов).

4. Обязательное условие начала промышленного производства — хронологическая сопоставимость оцениваемого образца и выбранного аналога по массовости производства и устойчивости спроса у потребителей. Недопустимо выбирать для сравнения аналоги, производи-

мые большими сериями, с образцами единичного изготовления, а также показатели, публикуемые в неофициальных (нефирменных) источниках.

5. Номенклатура показателей должна соответствовать действующим НТД или директивным документам. Базовые значения показателей технического уровня и качества оцениваемого СВТ выбирает главный конструктор изделия по согласованию: а) с межотраслевой головной организацией по группам однородных СВТ, б) с ГКВТИ СССР и межотраслевой головной организацией по системам однородных СВТ по принадлежности СВТ. При отсутствии директивно или нормативно установленной номенклатуры показателей технического уровня и качества ее определяет главный конструктор изделия по согласованию с межотраслевой головной организацией по группам однородных СВТ, ГКВТИ СССР и межотраслевой головной организацией по системам однородных СВТ по принадлежности СВТ.

6. Для каждой группы однородных СВТ номенклатура и приоритеты показателей технического уровня и качества разрабатываются межотраслевыми головными организациями по группам однородных СВТ и межотраслевыми головными организациями по системам однородных СВТ так, чтобы совокупность выбранных показателей наиболее полно характеризовала свойство данной группы и ее технический уровень и качество. Номенклатура показателей технического уровня и качества групп однородных СВТ и приоритеты показателей согласовываются с межотраслевыми головными организациями по системам однородных СВТ по принадлежности СВТ, утверждаются ГКВТИ СССР и издаются в виде директивных или нормативных документов. Отнесение СВТ к той или иной группе однородной продукции проводится на основании Классификатора СВТ, утверждаемого ГКВТИ СССР по согласованию с межотраслевыми головными организациями по системам и группам однородных СВТ.

7. Допускается в обоснованных случаях по инициативе главного конструктора изделия или заказчика (ГКВТИ СССР), по согласованию с ГКВТИ СССР и межотраслевыми организациями по системам и группам однородных СВТ расширять установленную номенклатуру показателей технического уровня и качества и изменять их приоритеты.

8. В качестве базового значения показателя технического уровня и качества СВТ можно выбрать либо его конкретное количественное значение, либо диапазон значений.

Если реальные образцы или информация о ряде их показателей отсутствуют, то в качестве значений (диапазонов значений) базовых показателей выбираются их прогнозируемые значе-

ния на интервал времени сбыта готовой продукции.

9. В качестве значений базовых показателей выбираются:

а) на этапах разработки технического задания, защиты эскизного и технического (эскизно-технического) проектов — значения (диапазоны значений), полученные методом прогнозирования и характеризующие мировой уровень техники в данной группе однородных СВТ на момент окончания разработки и начала ее серийного выпуска; значения показателей, установленные в НТД или в технических требованиях заказчика; при отсутствии прогнозных значений показателей и значений, установленных в НТД.

Если прогнозные значения показателей лучше, чем значения, регламентированные НТД или требованиями заказчика, то в качестве базовых выбираются прогнозные значения;

б) на этапе приемочных испытаний и на стадии серийного производства — значения (диапазоны значений) показателей лучших реальных образцов данной группы однородных СВТ,

пользующихся во время проведения оценки устойчивым спросом на мировом рынке.

10. Результаты оценки технического уровня и качества СВТ оформляют в виде карты технического уровня и качества продукции, которую включают в состав документации, сопровождающей техническое задание на разработку изделия и пояснительную записку к эскизному, техническому (эскизно-техническому) проектам. Других документов для характеристики технического уровня и качества СВТ не требуется.

11. При оценивании технического уровня и качества или при экспертизе принятых оценок по всем возникающим разногласиям принципиального характера решение принимает ГКВТИ СССР.

Временную методику оценки технического уровня и качества СВТ общего назначения можно заказать и получить в Управлении делами ГКВТИ СССР.

Телефон 252-21-85, Москва

УДК 621.865.8:681.3.06

В. С. Горбачев, В. Л. Попов

КРОСС-СИСТЕМА АУТОНОМНОГО ПРОГРАММИРОВАНИЯ ДЛЯ РОБОТА РМ-01 С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ ТРЕХМЕРНОЙ МАШИНОЙ ГРАФИКИ

Кросс-система автономного программирования робота РМ-01 (КАПР) предназначена для автономной разработки, создания и отладки управляющих программ для робототехнических комплексов (РТК) с промышленными роботами РМ-01. КАПР позволяет:

автономно определять и корректировать точки позиционирования ПР, эффективно обнаруживать и исправлять грамматические, алгоритмические и геометрические ошибки в управляющей программе;

тестировать программы с использованием средств трехмерной машинной графики, наблюдая движение манипулятора на графическом дисплее;

переносить автономно отлаженную управляющую программу в устройство управления СФЕРА-36 через линию связи или гибкий магнитный диск;

решать задачи компоновки РТК и размещения оборудования (создавать модели оборудования, перемещать их в пространстве, менять точку обзора, масштаб и т. д.).

Аппаратные средства системы базируются на ЭВМ СМ4, оснащенной графическим растровым дисплеем Электроника МС7401.

Программное обеспечение состоит из следующих компонентов:

отладчика для языка программирования ARPS, предоставляющего пользователю развитие средства контроля правильности выполнения программы;

подсистемы графического трехмерного моделирования, используемой без автономного программирования;

модуля определения и редактирования точек.

Предлагаемую систему можно поставить на ПЭВМ типа ЕС1841, «Искра 1030.11», «Электроника 85».

Телефон 289-51-75, Москва; адрес 220093, г. Минск, ул. Притыцкого, 65, НПО «Гранат».

КРАТКИЕ СООБЩЕНИЯ

УДК 681.324.01

Г. К. Круг, О. Н. Антипов, В. А. Зудин, Е. Б. Шитарев
ЛОКАЛЬНАЯ СЕТЬ «ЭЛЬФ» НА БАЗЕ ЭВМ

Локальная сеть «Эльф» разработана для автоматического управления распределенными системами. Она состоит из центральной ЭВМ с интерфейсом «Стык С2» или «20 мА токовая петля» и 16 модулей управления, удаленных друг от друга на расстояние до 150 м. Обмен информацией производится асинхронно со скоростью 1200...9600 Бод по последовательному каналу.

Модуль управления реализован на базе ЭВМ КМ1816ВЕ35 и имеет 2 Кбайт ПЗУ команд, 2 Кбайт ОЗУ команд, 4 Кбайт ОЗУ данных, 8-разрядный ЦАП, 4-канальный 8-разрядный АЦП. Он может работать как в автономном режиме, так и под управлением центральной ЭВМ, обеспечивающей инициализацию сети, оперативную смену программ в модулях, независимый опрос АЦП и задание режимов ЦАП модулей, управление обменом информацией в сети. Возможно увеличение вычислительной мощности отдельных модулей управления с помощью подключения ЭВМ с интерфейсом ИРПР. Для отладки загруженных программ используется пошаговый режим с индикацией адресов и данных.

Объединение нескольких локальных сетей в глобальную не требует дополнительных аппаратных затрат.

Подобные структуры могут применяться для расширения систем автоматизации научных и экспериментальных исследований, для построения гибких автоматизированных производств, в информационно-измерительных системах, а также в качестве встроенных интеллектуальных средств автоматизации с перепрограммируемыми функциями.

105835, ГСП, Москва, Е—250, ул. Красноказарменная, 14, МЭИ, кафедра автоматики, тел.: 362-75-23

Сообщение поступило 2.03.88

УДК 681.3.06

П. В. Полянский

ОРГАНИЗАЦИЯ ПРИКЛАДНЫХ БАЗ ДАННЫХ НА ПЭВМ С МАЛОЙ ЕМКОСТЬЮ ОЗУ

Одно из наиболее популярных применений ПЭВМ в мире — создание на их основе прикладных баз данных. Базы данных (БД) находят одинаково успешное применение в любой профессиональной деятельности, спрос на них растет и в случаях чисто домашнего использования ПЭВМ.

Для бытового применения практически доступны только две машины: «Электроника БК-0010» и «Микроша». Малый объем ОЗУ, применение языков интерпретирующего типа, отсутствие накопителей на магнитных дисках резко ограничивают возможность использования этих машин для организации БД. Многие специалисты считают, что на ПЭВМ такого класса организация БД вообще нецелесообразна.

Причины неэффективности БД на бытовых ПЭВМ. Организационные. Часто считают, что БД для ПЭВМ должны организовываться так же, как для больших ЭВМ, т. е. в памяти ПЭВМ должна храниться и быть доступной вся информация, которая может потребоваться пользователю. Следствие этого — *технические* причины. Интерпретаторы крайне неэкономно расходуют и без того малый объем ОЗУ, скорость их работы также не удовлетворяет данной задаче. БД, реализованные в таких условиях, становятся не мощным инструментом, а иллюстративным материалом. Программирование на ассемблере позволяет эффективно использовать память и быстродействие машины, но резко ограничивает возможности пользователя по модификации БД под свою конкретную задачу.

Особенности организации прикладных БД. Решить существующую проблему можно только при комплексном подходе. Сначала рассмотрим организационную сторону. При работе с неавтоматизированными БД (обычно это карточки) выделяются следующие наиболее трудоемкие операции, полностью определяющие эффективность этих БД: поиск информации по известным признакам и внесение закономерных изменений (например, перенумерация) внутри большого массива данных и действия на основе

полученной информации (составление и пересылка нового документа и т. д.).

Подавляющее большинство прикладных БД отличается высокой регулярностью и структурированностью данных, поэтому нет необходимости хранить в ПЭВМ всю имеющуюся информацию. Достаточно создать необходимый набор информационных признаков и хранить их в памяти машины вместе со ссылками на внешние источники, содержащие полную информацию. Такой подход резко снижает требования к объему ОЗУ, повышает скорость поиска. Возрастает и надежность, поскольку в зону поиска не попадает малозначимая информация. Облегчается ввод информации в БД, автоматизировать который на данном уровне пока слишком дорого.

В памяти ПЭВМ без значительных затрат можно разместить и поля для внесения оперативных пометок в виде ссылок на ограниченный словарь, также находящийся в памяти машины. Это позволяет указать степень важности информации и провести ее дополнительное группирование уже по локальным признакам. При этом сохраняется удобство восприятия данных, так как они представлены в виде текстовых сообщений на естественном языке.

Указанный подход в организации машинного варианта БД значительно упрощает логику проведения закономерных изменений в БД, что облегчает их программирование.

Технические аспекты реализации. Чтобы обеспечить высокие показатели ПО по скорости работы и эффективности использования ОЗУ при сохранении универсальности применения, ПО необходимо функционально разделить на две составляющие:

первая часть ПО — набор программ на ассемблере, реализующих основные функции БД (управление распределением памяти ПЭВМ; запись, считывание и удаление произвольной информации из БД; поиск информации по полному или сокращенному набору информативных признаков). Она в основном определяет скоростные и емкостные возможности системы;

вторая часть ПО — обеспечение решения БД конкретных прикладных задач (весь диалог с оператором, оформление экрана, реализация конкретных алгоритмов функционирования БД). Такое ПО необходимо писать на языке высокого уровня.

В ПЭВМ «Электроника БК-0010» обе части

ПО успешно объединяются в одну программу при использовании языка программирования БЕЙСИК [1]. При этом первая часть ПО существует в виде набора специализированных функций пользователя (USR) и таких директив языка, как CLFAR, ASC, CHR, STRING, позволяющих управлять распределением памяти в БЕЙСИК-системе и работать с символьной информацией.

Что касается языка Фокал, которым оснащена значительная часть машин «Электроника БК-0010», то он менее отвечает поставленной задаче. Но и здесь в рамках модернизированного интерфейса внешних функций [2] удалось создать систему внешних функций ПОИСК, полностью реализующих изложенный подход к организации БД.

Функции, задействованные в системе:

F.D<IMENSION>(D):D — количество данных (признаков) в одной карточке (ее длина). Устанавливает количество данных в карточке, очищает ОЗУ под данные системы;

F.W<RITE>(NK, P0, ..., PN): записывает в карту с номером NK список данных P0, ..., PN. Если NK=0, то записывается модель поиска для функции F.S;

F.R<EAD>(NK, 1): считывает значение 1-го данного карточки с номером NK;

F.E<RASE>(NK): стирает карточку с номером NK, уничтожает номер и все данные карточки, освобождает место под новую запись;

F.S<EARCH>: осуществляет последовательный поиск карточек, данные которых совпадают с моделью, введенной через F.W;

F.L<IBRARX>(DL, <имя файла>): обеспечивает запись и считывание с МЛ всей области памяти, где фактически содержатся данные системы.

Практические рекомендации. Рассмотрим возможную организацию телефонного справочника по магазинам Мосхозторга (их около 150) [3]. Можно предложить следующий состав информационных признаков: номер магазина; телефон; состав отделов; ссылка на страницу телефонного справочника (там часто несколько телефонов и точный адрес); номер квадрата карты города, где расположен магазин; ближайшая станция метро; вид и номер одного маршрута наземного транспорта; ориентировочное время в пути; дополнительные оперативные сведения (цитаты из ограниченного словаря, вводимые пользователем в процессе поиска).

Эта практически исчерпывающая информация о магазинах будет занимать в памяти ПЭВМ объем, не превышающий 3 Кбайт (20 байт×150). (При использовании только интерпретатора — 12 Кбайт.) Тексты названий всех станций метро (их почти 130) расположены в интерпретаторе в строках с операторами

PRINT (TYPE) и занимают около 3 Кбайт. Их наличие позволяет не отвлекаться на дополнительную расшифровку информации, полученной из БД.

В указанной БД можно не только предварительно отобрать нужные магазины, но и автоматически набрать их телефоны [4] с полуавтоматической коррекцией списка по результатам звонка. То есть в данном случае удастся автоматизировать не только получение нужной информации из БД, но и действия на ее основе.

Полностью в соответствии с изложенным подходом в рамках системы ПОИСК создана и эксплуатируется БД научной информации. В системе используется девять признаков, каждый из которых можно выразить числом в диапазоне 0...255 (1 байт), и шесть признаков с бинарным значением. ПЭВМ «Электроника БК-0010» запоминает в ОЗУ около 1000 подобных «карточек». Экранная память при этом сохраняется полностью, но абсолютное большинство текстовых ссылок приходится хранить на бумаге.

Этот недостаток во многом компенсируется высоким быстродействием и удобством работы с системой. Поиск по полной модели занимает менее 0,3 с. Общаются с системой на основе богатого набора различных меню, что позволяет работать с ней людям, имеющим минимальную подготовку в области ЭВМ.

Предлагаемый подход к организации БД полностью применим и к любым другим ПЭВМ, включая «Микрошу», которая в таком использовании имеет даже преимущество в емкости ОЗУ перед БК-0010. В «Микроше» все программы, включая БЕЙСИК, загружаются в ОЗУ общей емкостью 32 Кбайт. Экрану (не имеющему графических возможностей) отводится около 2 Кбайт. Используя минимальные версии БЕЙСИКа, под прикладную программу БД и собственно данные можно отвести примерно 23 Кбайт.

107005, Москва, 2-я Бауманская ул., д. 5, МВТУ
им. Н. Э. Баумана, каф. МТ-7, Полянский П. В.

ЛИТЕРАТУРА

1. Программное обеспечение микроЭВМ «Электроника БК-0010 «Электроника БК-0010Ш». Язык БЕЙСИК. Описание языка. 00001—01.35.02, 1987.
2. Казанцев А. П. Интерфейс внешних функций // Микропроцессорные средства и системы.— 1987.— № 4.— С. 86.
3. Список абонентов московской городской телефонной сети. Телефоны организаций, учреждений и предприятий.— М.: Радио и связь, 1982.
4. Ярошенко В. Программа «телефон» // Наука и жизнь.— 1987.— № 10.— С. 104.

Статья поступила 24.11.87

РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ СУБД КВАНТ

Персональные ЭВМ типа ДВК находят все большее применение в системах обработки данных, но небольшой объем оперативной памяти (32 Кслов) в моделях ДВК2, ДВК3, относительно большое время доступа к накопителям на гибких магнитных дисках и небольшой объем внешней памяти — это серьезное препятствие при построении на базе ДВК систем обработки данных с большими объемами информации.

В связи с этим многообещающим является направление, получающее все более широкое распространение, по использованию ДВК в распределенных системах обработки данных [1]. Большинство разработанных программных средств содержит в качестве элементов распределенной обработки лишь режим эмуляции терминала и пересылку текстовых файлов между центральной и сателлитной ЭВМ и не обеспечивает непосредственное взаимодействие программы, работающей в сателлитной ЭВМ с СУБД центральной ЭВМ.

Рассмотрен аппаратно-программный комплекс для распределенной обработки, реализующий возможность обращения к СУБД КВАНТ, функционирующей на центральной ЭВМ, из прикладной программы, функционирующей на ДВК. При этом ДВК выполняет, в основном, функцию взаимодействия с пользователем, а на центральную ЭВМ возложены функции, связанные с обращением к базе данных (БД). В качестве центральной используется ЭВМ «Электроника 100/25» (а также «Электроника 79») под управлением ОС РВ [2] с СУБД КВАНТ [3], к которой подключен ряд ДВК, работающих в ОС РАФОС [4].

В комплексе для распределенной обработки данных (рис. 1) ЭВМ «Электроника 100/25» подключена к линии связи через мультиплек-

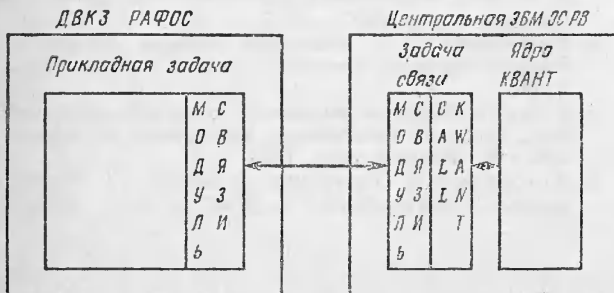


Рис. 1. Структурная схема комплекса технических средств

сор МПА-1 (15КС16-043), а ДВК3 — через блок ИРПС, обеспечивающий работу бытового компьютера «Электроника БК-0010Ш» в составе КУВТ [5]. К МПА-1 блок ИРПС подключается посредством четырехпроводной линии связи. ДВК3 удален от ЭВМ «Электроника 100/25» на 50 м. Скорость обмена по каналу «токовая петля» составляет 9600 Бод.

СУБД КВАНТ — мощная система с развитыми возможностями обработки данных, недостатком которой является значительный объем используемой оперативной памяти. Для ЭВМ «Электроника 100/25» (или «Электроника 79») с объемом ОЗУ 256 Кбайт это приводит к тому, что кроме ядра СУБД КВАНТ одновременно в памяти ЭВМ могут находиться одна-две прикладные задачи, работающие с СУБД. При увеличении числа задач ОС организует свопинг, что делает неприемлемым время реакции в интерактивном режиме работы пользователя.

При нераспределенной обработке прикладная программа, работающая с СУБД КВАНТ, осуществляет обращение к БД посредством вызова специальной процедуры, которая передает вызов ядру СУБД КВАНТ, обычно резидентному в ОЗУ. В качестве параметров из прикладной программы передаются три массива: управляющий блок, буферы записи и поиска.

Управляющий блок содержит код операции, имя подсхемы и другую информацию, необходимую ядру КВАНТ для обработки запроса. Буфер записи обычно имеет информацию, которую нужно занести в БД, или информацию, полученную из БД в случае, если прикладная программа выдала запрос на обращение к БД. Буфер поиска содержит поисковый запрос в виде отношения между дескрипторами и используется для выбора из БД записей, удовлетворяющих этому отношению.

В описываемом комплексе ядро СУБД КВАНТ является доминирующей задачей ЭВМ «Электроника 100/25» в смысле оперативной памяти и процессора. Все прикладные задачи переносятся в ДВК. Вместо них в центральной ЭВМ остаются только небольшие задачи связи с ДВК, число которых может быть порядка одного-двух десятков без свопинга.

В структурной схеме (рис. 2) задача связи в центральной ЭВМ имеет имя ADADVК. ДВК рассматривается как терминал и обмен данными с ней осуществляется через терминальный драйвер обращением к директиве QIO с подавлением интерпретации управляющих символов. Задача связи в ДВК имеет имя KWANT. Обмен данными с центральной ЭВМ производится через регистры данных и состояния блока ИРПС. Предусмотрено обращение к задаче из программ на языках

Фортран и Паскаль. В качестве параметров обращения к задаче связи используются три массива: управляющий блок, буферы записи и поиска.

Таким образом, обращение к модулю KWANT из прикладной программы на ДВК имеет точно такую же форму, как обращение к СУБД КВАНТ в любой прикладной программе центральной ЭВМ, работающей под управлением ОС РВ. Благодаря этому программа, разработанная для ОС РВ на ЭВМ «Электроника 100/25», может быть перенесена в виде исходного текста в ДВК без каких-либо изменений, а затем оттранслирована и построена в ОС РАФОС, при этом на этапе редактирования внешних связей вместо объектной библиотеки СУБД КВАНТ используется одноименный объектный модуль связи.

Протокол обмена между задачами связи ДВК и центральной ЭВМ сводится к процедуре последовательной передачи параметров обращения из прикладной программы через задачу связи KWANT к задаче связи ADADVK центральной ЭВМ, обращения к ядру СУБД КВАНТ из задачи связи, после которого значение параметров может измениться, и обратной передачи параметров в ДВК прикладной программе через задачу связи KWANT. В контрольных задачах время одного обращения к СУБД центральной ЭВМ из прикладной программы, функционирующей в ДВК, составило 1...3 с, что приемлемо для интерактивного режима работы.

Наиболее эффективно рассматриваемая система может применяться в режимах экранного редактирования БД центральной ЭВМ пользователем, работающим за терминалом ДВК, например для ввода и контроля данных. При этом все функции экранного взаимодействия с пользователем и контроля информации осуществляются программой в ДВК, а центральная ЭВМ используется лишь в моменты, когда необходимо поместить готовую запись в БД центральной ЭВМ. Реактивность системы повышается, если при передаче управляющего блока выбирать только необходимые его элементы.

Телефон 532-88-04, Москва

ЛИТЕРАТУРА

1. Вейцман К. Распределенные системы мини- и микроЭВМ.— М.: Финансы и статистика, 1982.— 382 с.
2. Операционная система ОС РВ СМ ЭВМ: Справочное издание/Т. А. Егоров, В. Л. Кароль, И. С. Моствов и др.— М.: Финансы и статистика, 1987.
3. Государственная система промышленных приборов и средств автоматизации: Система малых ЭВМ. Программное обеспечение.— М.: ЦНИИТЭИприборостроения, 1984.— Т. 5, вып. 4.— С. 28—30, 32—33.
4. Операционная система СМ ЭВМ РАФОС. Спра-

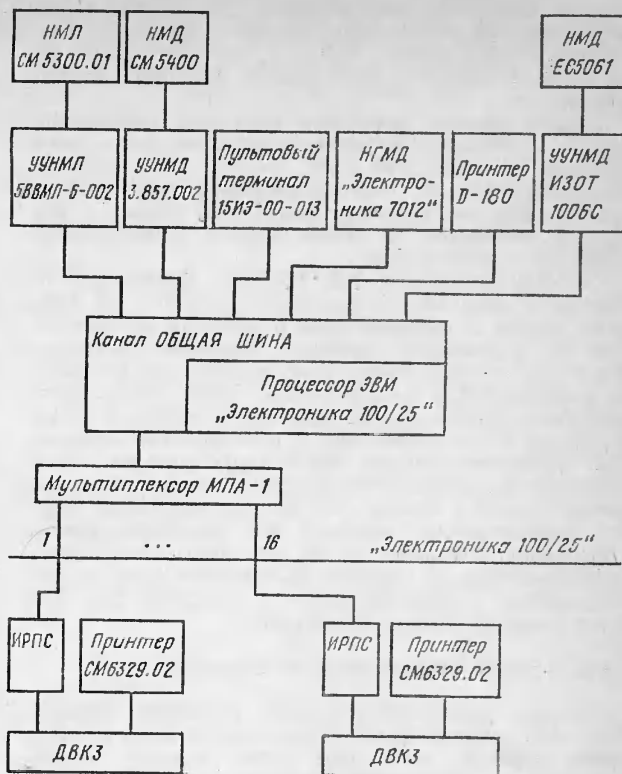


Рис. 2. Схема взаимодействия программного обеспечения ДВК и центральной ЭВМ

- вочник / Под. общ. ред. В. П. Семика.— М.: Финансы и статистика, 1986.— 208 с.
5. Комплексный класс технических средств на базе микроЭВМ «Электроника БК-0010Ш» и ДВК 2МШ/Г. И. Фролов, С. М. Косенков, В. А. Шахнов и др. // Микропроцессорные средства и системы.— 1986.— № 4.— С. 65—66.
 6. Одноплатные микроЭВМ ряда «Электроника МС 1201»/В. Л. Дшхунян, Ю. И. Борщенко, Ю. Л. Отроков, С. А. Шишарин // Микропроцессорные средства и системы.— 1985.— № 2.— С. 8—13.
 7. Мини- и микроЭВМ семейства «Электроника»/Б. Л. Толстых, И. Л. Талов, В. Г. Цывинский и др.— М.: Радио и связь, 1987.

Статья поступила 8.04.88

УДК 658.012.011.56

С. В. Казменко

БАЗА ДАННЫХ ПРИ ИНТЕГРАЦИИ ДИАЛОГОВЫХ СРЕДСТВ КОНЕЧНОГО ПОЛЬЗОВАТЕЛЯ

Введение

Системы интегрированного ПО обычно классифицируют по их архитектуре и месту в ПО ЭВМ. В настоящей статье предполагается классификация систем по другому признаку — среде интеграции. Для большинства систем средой интеграции является либо крупноформатное рабочее поле (простыня) или несколько таких простыней, либо иерархическое множество фреймов [1—4]. Еще одно решение проблемы среды — исполь-

зование для этих целей экранных документов, например в системе Office — by — example (OBE) [5]. Рассмотрим базу данных (БД) в качестве среды интеграции диалоговых возможностей конечного пользователя.

Каждая из трех выделенных концепций («простыня», фреймы и экранные документы) позволяет организовать средства взаимодействия с БД. Вместе с тем, экранные документы наиболее понятны широкому кругу пользователей. Все это побудило рассмотреть вопрос о возможной реализации на основе средств взаимодействия с БД среды пользователя.

Системы управления БД (СУБД), работающие на ПЭВМ, не обеспечивают диалоговую работу с БД большого объема и ориентированы в основном на простейшие БД реляционной структуры. Возникает противоречие между объемом информации, необходимой для работы пользователя, и возможностями ПЭВМ. Это противоречие разрешается за счет введения естественной структуризации БД в соответствии с особенностями отражаемой предметной области. Такая структуризация может осуществляться посредством «устаревших» моделей данных (иерархической и сетевой), для которых необходимы меньшие вычислительные мощности при одинаковом объеме информации. Считается [6, 7], что нельзя реализовать ориентированные на конечного пользователя средства взаимодействия с иерархическими и сетевыми БД. Как будет показано дальше, это не так.

Базы данных в качестве среды пользователя

Системы управления БД часто выполняют функции среды пользователя. При этом программная среда строится таким образом, что СУБД имеет мощную систему интерфейсов с несколькими языками различных уровней. Как отмечается в [8], интерфейсы СУБД с включающими языками значительно определяют потенциальные возможности системы при разработке нестандартных приложений. Логическим развитием программного окружения БД являются внутренние языки СУБД, которые можно назвать командными языками запросов.

Большое значение работе конечного пользователя с БД уделяется в публикациях, посвященных общим вопросам построения диалога [9—12] и диалоговым системам общего назначения [5, 10, 11, 13—16]. Анализ предлагаемых способов организации диалога показывает, что, с одной стороны, диалог в различных работающих с БД системах имеет много общего, а с другой — каждый разработчик при проектировании и реализации своей системы выступает каквольный художник. Это объясняется тем, что разработка дружественного пользователю интерфейса является в настоящее время искусством, а не наукой (an art and not a science) [17].

При создании средств доступа конечного пользователя, несмотря на все показанные сложности, к БД предъявляются повышенные требования к комфортности диалога. Наилучшим образом удовлетворить эти требования можно, применив принцип непосредственного манипулирования [11] (в терминологии работ [9, 10] — непосредственного редактирования).

По сравнению с теми, вполне очевидными, преимуществами, которые предоставляет основанное на непосредственном манипулировании данными взаимодействие человека с ЭВМ, преимущества основного альтернативного способа — взаимодействия с ЭВМ на естественном языке [18], представляются весьма проблематичными. Развернутая критика взаимодействия с ЭВМ на естественном языке приведена в работе [19].

Особенности применения принципа непосредственного манипулирования при работе с БД

Взаимодействие с БД на основе принципа непосредственного манипулирования можно обеспечить с помощью механизма форматирования экрана. Исторически раньше

получили широкое распространение системы форматного ввода информации в ЭВМ [20]. Для наиболее совершенных образцов этих систем использование принципа непосредственного манипулирования состоит в следующем: присутствие на экране некоторого бланка (формата, пустографки и т. д.); для каждого элемента данных наличие на экране своего окна, в котором пользователь вводит конкретное значение элемента; использование набора функциональных клавиш.

Применение принципа непосредственного манипулирования при взаимодействии с БД можно организовать аналогично: наличие на экране бланка; отображение элементов данных в графах бланка; использование функциональной клавиатуры. При взаимодействии с БД целесообразно выделить две основные ситуации: работа пользователя с отдельной записью БД и с некоторой совокупностью записей. Еще один важный момент состоит в том, что при организации диалога пользователя с БД очень удобным средством являются таблицы. Практика разработки систем, реализующих табличный способ взаимодействия с БД [10, 21—23], подтверждает это.

Внешняя модель данных

Традиционно внешняя модель данных ориентирована на программиста [24—26]. Непосредственное участие непрограммирующих специалистов в автоматизированном решении задач с помощью ЭВМ делает необходимой ориентацию внешней модели данных непосредственно на конечного пользователя.* При этом необходима такая модель, которая будет совпадать с привычными представлениями пользователя.

Для обычных бумажных документов характерны следующие особенности:

наличие простой структуры: документ есть некоторая совокупность составляющих его элементов (реквизитов); наличие такой части документа, которая состоит из строк определенной структуры;

включение в документ наряду со структурированными элементами данных (числами, шифрами, кодами, символическими переменными фиксированной длины и т. д.) произвольного текста, который представляется как некоторая последовательность строк.

Рассмотрим взаимосвязь внешней модели с внутренней. На внутреннем уровне для БД различной структуры (иерархической, сетевой, реляционной) может быть определено такое понятие, как состоящая из реквизитов запись БД. Например, для реляционной БД в качестве такой записи может выступать кортеж, а в качестве реквизитов — входящие в кортеж атрибуты. В иерархических и сетевых БД запись может соответствовать сегменту базы. Даже в случае, когда физически записей как таковых в базе нет, они могут быть специально построены (логические записи) с целью организации взаимодействия пользователя с БД.

С учетом первой особенности бумажных документов при построении внешней модели можно предположить, что документ определенной структуры с фиксированным числом реквизитов соответствует логической записи БД. Такую запись целесообразно предоставлять пользователю в виде экранного документа (ЭД). Таким образом, ЭД — частный случай представления внешней модели. Далее будем считать понятие внешней модели тождественным некоторой совокупности экранных документов, хотя внешняя модель — более общее понятие и может выражаться в других формах.

Последовательность записей является столь же простой для понимания широкого круга пользователей информационной структурой, как и запись, состоящая из

* В общем случае внешняя модель должна быть ориентирована на конечных пользователей и программистов, однако в дальнейшем рассматриваются только аспекты использования внешней модели, не связанные с программированием.

отдельных реквизитов, что соответствует второй отмеченной особенности бумажных документов (наличию в документах списка структурированных строк). Так, целесообразна организация ЭД с последовательностью строк, каждая из которых состоит из некоторого набора показателей одной внешней записи БД. Таким образом, во внешней модели можно учесть необходимость доступа конечного пользователя не к отдельной записи БД, а к некоторой их совокупности (последовательности). Такие последовательности, как правило, физически присутствуют в БД с любой внутренней моделью данных.

Единообразное внешнее представление БД в виде записей и их последовательностей позволяет говорить о единой внешней модели, а следовательно, в едином подходе к взаимодействию с БД и, в частности, о стандартизации диалога для БД, построенных на основе различных моделей.

Предложенная внешняя модель данных может быть использована для диалоговой интеграции неоднородных БД. Особенности унифицированной модели данных для неоднородных БД рассмотрены в [25, 27], а вот вопрос о удобных, ориентированных на конечного пользователя, унифицированных средствах взаимодействия с различными БД является недостаточно проработанным.

Внешние модели для организации взаимодействия пользователя с БД

Представляя внешнюю модель БД в виде множества экранных документов, целесообразно различать ЭД двух видов: экранные формы и таблицы. При этом формы будут использоваться для представления отдельных логических записей БД, таблицы — для последовательностей записей. Соединяя ЭД двух разных видов, получаем документ сложной структуры, имеющий заголовок в виде формы и последовательность строк в виде таблицы. Например, заголовок может содержать элементы данных некоторой логической записи базы, а строки ЭД — являться «срезами» последовательности подчиненных записей. Другой пример: заголовок содержит атрибуты, одинаковые для всех кортежей отношения, а в строках представлены переменные части кортежей. Целесообразность выделения таких двухуровневых документов отмечается, в частности в [28].

Вне зависимости от внутренней модели данных при работе пользователя с внешними (логическими) записями базы одни и те же операции могут выполняться одинаково. Так, при форматном редактировании отдельной записи функциональные клавиши, в частности клавиши перехода к следующему (предыдущему) реквизиту, могут означать одно и то же действие вне зависимости от внутренней модели данных. Работа пользователя с последовательностью записей может быть реализована в виде диалоговой работы с таблицей.

Взаимодействие пользователя с БД на основе предлагаемой внешней модели можно организовать в форматном и табличном режимах.

В форматном режиме (Ф-режим) пользователю доступны для обработки (в частности, просмотра) все элементы данных одной записи. Если число элементов данных в записи некоторого типа (записи одного типа, которые имеют одинаковую структуру) велико и все элементы не могут быть одновременно представлены на экране, то при описании схемы БД визуальное представление записи делит на кадры (первый, второй и т. д.).

В табличном режиме (Т-режим) пользователь работает только с некоторыми элементами, зато имеет визуальный доступ ко многим записям таблицы.

БД для интеграции диалоговых средств конечного пользователя

Важной предпосылкой для обеспечения единства доступа к данным различного вида является признание целесообразности хранения всех видов данных в одних

и тех же БД [30]. При этом обеспечение конечному пользователю удобного взгляда на такую интегрированную БД, а также предоставление ему возможности (посредством реализации определенной технологии) формировать и изменять содержимое БД, работая с экранными документами, означало бы создание интегрированной среды конечного пользователя.

Все подлежащие интеграции компоненты ПО можно разбить на два класса: специализированные — те, которые ориентированы на обработку своего специфического вида данных (обработка текста, машинная графика, обработка звуковых сообщений и др.); функциональные (сетевая передача данных, вычислительные и др.).

Специализированные компоненты интегрируются включением специфических элементов данных в логические записи БД. При этом соответствующие диалоговые средства конечного пользователя будут иницироваться в нужный момент. Например, логическая запись БД может отражаться на экран посредством перебора всех элементов конкретной записи. Каждый элемент данных обрабатывается своим компонентом (своим модулем), зависящим от вида элемента. Аналогично: если некоторый элемент данных является графическим, то вызывается программный модуль построения графического образа.

Динамические элементы данных (речеграммы, мультфильмы и т. д.) могут при воспроизведении на экранном документе изображаться визуальной меткой, например «речеграмма — аннотация документа». Для воспроизведения динамических элементов данных в любой момент целесообразно иметь специальные команды.

Интеграцию функциональных компонентов рассмотреть сложнее, так как у них нет единого способа включения в интегрированную систему. Каждый функциональный компонент необходимо рассматривать отдельно.

Особое место занимает компонент обмена. После отправки (адресования) сообщения или документа пользователь, как правило, не имеет возможности следить за его прохождением. Таким образом, компонент сетевого обмена как бы отдален от пользователя.

БД в качестве среды интеграции диалоговых средств обработки текста и структурированных данных

Средства обработки текста на универсальной ЭВМ реализуются в виде работающего только с текстом редактора или процессора [32]. В результате недостаточно используется возможность совместной обработки текста и данных. Даже в интегрированных системах, которые «по определению» должны бы предоставлять развитые возможности такой совместной обработки, как отмечается в [1], текстовый процессор, как правило, не связан с данными, обрабатываемыми с помощью электронных таблиц и СУБД.

Некоторые интегрированные системы ПО позволяют организовать совместную обработку текста и структурированных данных. Однако при этом средой интеграции является крупноформатное рабочее поле или фрейм, что делает такие системы непривычными для пользователей-специалистов по обработке данных. Для таких пользователей при построении средств одновременной работы с текстом и данными необходимо учитывать третью (из отмеченных при рассмотрении внешней модели БД) особенность обычных бумажных документов — совмещение в них структурированных данных и текстовых фрагментов. Необходимо, чтобы текстовые фрагменты являлись элементами данных и принадлежали в этом качестве (элементов) логическим записям БД. Возможно, что экранный документ как представление логической записи БД будет содержать несколько текстовых фрагментов.

Поддержка средств диалогового взаимодействия с БД при наличии в записях текстовых фрагментов может осуществляться с помощью алгоритмов, аналогичных приведенным в [33]. При этом единственное отличие: если очередной обрабатываемый элемент данных является

текстовым фрагментом, то вызывается модуль, реализующий возможности экранного редактора.

Работа с текстом в рамках ФТ-подхода целесообразна только в форматном режиме. Это обусловлено тем, что при расположении текстового фрагмента внутри представленной на экране строки таблицы маляя информационная емкость вступает в противоречие с целостностью восприятия документа пользователем: система будет показывать ему слишком малую часть текста в каждой логической записи БД.

ФТ-подход позволяет с помощью форматного представления решить диалоговый аспект интеграции обработки текста и данных и не затрагивает других аспектов больших документально-фактографических БД. Интеграция данных и текста в больших БД — крупная проблема. При ее решении, с одной стороны, имеется тенденция к раздельной реализации, так, даже при проектировании машин БД, как правило [34], осуществляется раздельная разработка машин для управления форматизированными и текстовыми БД. С другой стороны, часто специально разрабатываются интегрированные документально-фактографические системы [35—37].

Персональные вычислительные средства в среде БД

Обычно в системах интегрированного ПО в качестве средств персональных вычислений (СПВ) используются крупноформатные электронные таблицы (КЭТ), одновременно являющиеся и той средой, на основе которой осуществляется интеграция. Использование КЭТ в качестве среды персональных вычислений настолько популярно, что попытки предложить другую среду неизбежно встретят возражения. Поэтому цель настоящего раздела показать принципиальную возможность организации СПВ в среде БД. Это позволит сделать вывод о том, что среда БД — не только среда интеграции по данным, но и полнокровная интегрированная среда для конечного пользователя.

При использовании БД в качестве среды интеграции применение КЭТ для организации СПВ порождает проблемы, связанные с тем, что КЭТ — это еще одна дополнительная структура данных. Альтернативой является создание СПВ, работающих непосредственно в среде БД. При построении таких средств можно использовать «программирование методом примеров» [38, 39].

Рассмотрим конкретный пример использования СПВ в среде БД. При работе в Ф-режиме пользователь может перейти во вспомогательный режим персональных вычислений. В этом режиме ему доступны несколько представленных на экране числовых регистров, с которыми он может выполнить следующие действия: арифметические и другие операции, доступные при работе с обычным непрограммируемым калькулятором (ввод числа в регистр, пересылка записей из одного регистра в другой и т. д.); выборка некоторого реквизита из логической записи БД в регистр и обратная операция (перенос значения из регистра в БД). Режим персональных вычислений может иметь две модификации: с запоминанием инициируемой пользователем последовательности действий и без запоминания. В случае запоминания система формирует и хранит вычислительный макрос, который впоследствии может выполняться автоматически [40, 41].

Сравнение различных способов интеграции

С точки зрения той среды, на основе которой осуществляется интеграция компонентов ПО, можно выделить классы систем со средой интеграции в виде КЭТ, множества фреймов и экранных документов. Интеграция на основе БД — это фактически интеграция на основе ЭД, который является средством представления пользователю информации из БД.

Одно из наиболее распространенных подходов к интегрированному ПО — его создание на основе крупноформатного рабочего поля (SPREADSHEET, «простыня», КЭТ). Особенность такого подхода: БД находится вне

среды интеграции и возникает необходимость переноса информации из БД на рабочее поле (в КЭТ). Это создает неудобства при работе пользователя. В большинстве существующих систем интегрированного ПО такой перенос осуществляется вполне успешно за счет того, что в качестве БД в этих системах используются реляционные БД чрезвычайно простой (даже среди реляционных БД) структуры. Фактически БД в таких системах — это совокупность последовательных файлов, индексированных по нескольким полям.

Применение КЭТ в качестве среды интеграции приводит к тому, что очень большой задел, связанный с теорией и практикой БД, не используется. С таким положением можно мириться на начальном этапе развития интегрированных систем, в будущем же при использовании КЭТ в качестве среды интеграции неизбежно возникнет необходимость переноса информации в КЭТ из более сложных БД. При этом пользователю придется работать с двумя механизмами: КЭТ и средствами переноса информации из БД в КЭТ. Причем, средства переноса по мере перехода к все более сложным БД будут усложняться. В результате при работе пользователю придется часто переходить из КЭТ в БД и обратно. Такая работа окажется нетехнологичной для определенной категории конечных пользователей.

Использование в качестве среды интеграции системы фреймов очень удобно в случае разработки крупного документа, например книги или отчета. Однако применение в таких системах достаточно мощных БД еще сложнее по сравнению с КЭТ. Явное достоинство системы фреймов — единый взгляд на все виды данных, который характерен и для интеграции на основе форм (экранных документов), хотя и в меньшей степени: в этих системах большинство компонентов также могут объединяться на единой основе — на входящих в форму различных элементах данных (числовых, текстовых, графических и т. д.). Однако СПВ для системы фреймов представляется в виде КЭТ, «втиснутой» в фрейм, а в форматных системах СПВ не является типовой структурой.

Основные потенциальные пользователи систем, в которых интеграция осуществляется на основе экранных документов, — работники сферы организационного управления. Классической системой при этом является ОВЕ. Так как в ФТ-подходе среда интеграции — БД, представляющаяся пользователю в виде ЭД, можно считать, что имеется определенное сходство между предложенным ФТ-подходом и подходом, принятым в системе ОВЕ. Однако в системе ОВЕ не в полной мере оказывается реализованным принцип непосредственного манипулирования и визуальности. В ОВЕ некоторый объект (то, что в принятой здесь терминологии соответствует логической записи БД и представляется в виде ЭД) может иметь такое информационное содержание, что не будет предоставлен пользователю ни по одному из его обычных запросов. Такой объект будет «в виде мусора пылиться в дальнем углу информационной системы». В ФТ-подходе при визуальном просмотре таблицы пользователь видит все принадлежащие некоторой окрестности логические записи БД, что способствует его своевременным действиям по удалению ненужной информации.

Выводы

1. При непосредственной работе на ЭВМ конечного пользователя возникает противоречие между его потребностями в информации и диалоговыми возможностями СУБД, работающих на ПЭВМ.

2. Эти противоречия во многих случаях можно разрешить за счет естественного структурирования информации посредством представления ее в виде иерархических и сетевых структур.

3. Для непосредственной работы с различными по структуре БД предлагается унифицированная, ориентированная на конечного пользователя, модель данных, на основе ко-

торой можно реализовать средства взаимодействия конечного пользователя с произвольными БД.

4. Представленная пользователю в виде совокупности экранных документов БД может являться полной по своим функциональным возможностям интегрированной средой.

Телефон 3-32-82, Димитровград

ЛИТЕРАТУРА

1. Александров В. В., Чернышова Л. В. Интегрированное программное обеспечение (аналитический обзор) // УСИМ.— 1986.— № 4.— С. 8—15.
2. Гнездилова Г. Г. Интегрированная операционная среда персональной ЭВМ // Микропроцессорные средства и системы.— 1987.— № 1.— С. 31—34.
3. Мазурик В. П. Прикладные системы и решение задач // Персональные компьютеры: информатика для всех.— М.: Наука, 1987.— С. 102—142.
4. Брябрин В. М., Блинов Д. М. Интегрированная система для решения прикладных задач // Микропроцессорные средства и системы.— 1987.— № 1.
5. Zlotof M. M. Office — by — example: a business language that unifies data and word processing and electronic mail // IBM systems journal.— 1982.— Vol. 21, N 3.— P. 272—304.
6. Кокорсва Л. В., Малашин И. И. Проектирование банков данных.— М.: Наука, 1984.— 256 с.
7. Зайцев Н. Г. Критический анализ концепций построения баз данных // УСИМ.— 1983.— № 4.— С. 81.
8. Чередниченко А. А. Состояние и тенденции развития систем управления базами данных для персональных ЭВМ // Измерения, контроль, автоматизация (ИКА).— 1986.— Вып. 4 (60).— С. 53—60.
9. Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации // Микропроцессорные средства и системы.— 1986.— № 1.— С. 44—46, 58.
10. Практическое программирование, проектирование и разработка диалоговых систем. Не традиционный подход / А. Г. Кушниренко, Д. В. Варсанюфьев, А. Г. Дымченко и др.— М.: МГУ, 1985.— 89 с.
11. Schneiderman V. Direct manipulation: a step beyond programming languages // Computer.— 1983.— August.
12. Шнейдерман Б. Психология программирования: человеческие факторы в вычислительных и информационных системах. Пер. с англ.— М.: Радио и связь, 1984.— 304 с.
13. Подковыров А. И. Образное программирование для информационно-справочных систем АСУП // УСИМ.— 1983.— № 5.— С. 74—77.
14. Программно-технологическое средство интерактивного взаимодействия конечных пользователей с информационной базой — ППП ИКП/С. Ф. Манович, Г. П. Мандрусова, А. В. Максакос и др. // УСИМ.— 1985.— № 6.— С. 107—109.
15. Ries D. R., Lin W. K., Barnett J., Chilennes R. M. An architecture for a database centered office system (DACOS) // Journal of telecommunication networks.— 1983.— Vol. 2, N 4.— P. 445—462.
16. Kaufman C. W., Barnett J., Blaustein B. The forms-based query system // Journal of telecommunication networks.— 1983.— Vol. 2, N 4.— P. 465—482.
17. Thesen A., Beringer D. Goodness-of-fit in the user-computer interface: a hierarchical control framework related to «Friendliness» // IEEE trans. syst., man and cybern.— 1986.— Vol. 16, N 1.— P. 158—162.
18. Рыбаков Ф. И. Системы эффективного взаимодействия человека и ЭВМ.— М.: Радио и связь, 1985.
19. Казменко С. В. Использование естественного языка для общения с ЭВМ: мнение пессимиста // НТИ. Сер. 2.— 1987.— № 6.— С. 32.
20. Казменко С. В. Системы форматного ввода информации в ЭВМ // УСИМ.— 1987.— № 1.— С. 98—102.
21. Казменко С. В. Диалог в СУБД ПЛАТОН.— Препринт.— М.: ЦНИИАтоминформ, 1986.— 31 с. (НИИАР, № 19 (700)).
22. Казменко С. В. Форматно-табличный подход к организации диалога конечного пользователя с базами данных // Электронная техника. Сер. 9. Экономика и системы управления, вып. 3 (60).— 1986.— С. 50—54.
23. О реализации документального контура многоцелевой информационно-справочной системы/В. И. Мордань, Ф. Д. Коружин, Н. Н. Грунцин, В. И. Карпенко // УСИМ.— 1984.— № 1.— С. 93—98.
24. Мартин Д. ж. Организация баз данных в вычислительных системах.— М.: Мир, 1978.— 616 с.
25. Калининченко Л. А. Методы и средства интеграции неоднородных баз данных.— М.: Наука, 1983.— 424 с.
26. Дейт К. Введение в системы баз данных.— М.: Наука, 1980.
27. Калининченко Л. А. Интеграция неоднородных баз данных и унификация моделей данных разнотипных СУБД // Материалы III Всесоюз. конф. «Банки данных» (Таллинн, 24—26 сент. 1985 г.) — Таллинн: Политехн. ин-т, 1985.— С. 131—149.
28. Ершов А. М., Маркова Н. Ф., Сычов П. М. Метод ввода и обработки документов с использованием интерпретации описаний.— Препринт.— Дубна, 1985.— 7 с. (ОИЯИ, № р 10—85—686).
29. Казменко С. В. Взаимодействие конечного пользователя с базами данных на основе форматно-табличного подхода // УСИМ.— 1988.— № 2.— С. 67—71.
30. Burich T. G. Will this be age of systems integration? // Infostems.— 1986.— Vol. 33, N 5.— P. 38—40.
31. Кинг К. Дж., Марьянски Ф. Дж. Тенденция развития учрежденийских информационных систем: Пер. с англ. // ТИИЭР.— 1983.— Т. 71, № 4.— С. 83—95.
32. Casey D. Word processing within data processing // Data process.— 1983.— Vol. 25, N 10.— P. 15—18.
33. Казменко С. В. Программный контроль входной информации при вводе с терминала // Электронная техника. Сер. Экономика и системы управления.— 1983.— Вып. 3.— С. 51—52.
34. Калининченко Л. А., Курылев В. А., Рывкин В. М. Повышение производительности реляционных СУБД в машинах баз данных // Измерения, контроль, автоматизация.— 1987.— № 2(62).— С. 73—84.
35. Петрусь А. В., Харин Н. П. Интегрированная документально-фактографическая система на базе ИПС ИСКРА и СУБД ДИСОД // Проблемы создания и применения диалоговых информационных систем в автоматизированных системах организационного управления. Ч. 1/Сб. Под ред. В. И. Дракина, Б. С. Березкина, Е. И. Ломако.— Таллинн: Валгус, 1985.— С. 19—22.
36. Дракин В. И., Березкин Б. С., Петрусь А. В. Документальные информационные системы в организационном управлении // Проблемы создания и применения диалоговых информационных систем в автоматизированных системах организационного управления. Ч. 1/Сб. под ред. В. И. Дракина, Б. С. Березкина, Е. И. Ломако.— Таллинн: Валгус, 1985.— С. 12—18.
37. Муромский А. А., Туляков К. В. Организация интегрированной базы данных на основе сетевой СУБД.— М.: ВЦ АН СССР, 1986.— 23 с.
38. Локовски Ф. Х. Повышение производительности учрежденческого труда: технические перспективы: Пер. с англ. // ТИИЭР.— 1983.— Т. 71, № 4.— С. 73—82.
39. Halbert D. S. An example of programming by example // Dept. Elec. Eng. and Computer Sci., Univ. of California, Berkeley, CA, Tech. Rep, 1981.
40. Казменко С. В. Автоматизация проектирования систем на основе объединения концепций макетирования и программирования без программиста // Микропроцессорные средства и системы.— 1987.— № 4.
41. Казменко С. В. Способ построения диалоговых визуальных средств персональных вычислений.— Препринт, ЦНИИАтоминформ, 1988.— 8 с. (НИИАР, № 1 (732)).

Статья поступила 12.05.88

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ РЕАЛИЗАЦИЯ ДИАЛОГА ПРИ ДОСТУПЕ К РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ

В работе [1] сформулированы принципы диалога, основанного на непосредственном редактировании информации. Они обоснованы как апеллирующей к способам работы и взаимодействия полушарий головного мозга, так и опытом работы с динамичными игровыми программами, которым нельзя отказать в высокой эффективности взаимодействия с человеком.

Основное преимущество диалога в системах с непосредственным редактированием информации заключается, как это ни парадоксально, в его отсутствии в привычном смысле. Человек в этом случае перемещается в «Мире объектов» своей задачи и изучает, преобразует или создает их. Особая эффективность диалога связана с активным использованием в программах знаний о текущем положении в «мире». Это и приводит, как замечает автор [1], к очень важной и характерной замене традиционной схемы «команда — аргумент, команда — аргумент, ...» на схему «аргумент — команда, аргумент — команда, ...».

Непосредственное редактирование информации изменяет не только диалог, но и сам процесс его проектирования. Часть задачи, связанная с проектированием интерфейса с пользователем, казавшаяся тривиальной и скучной, превращается в творческий процесс создания «Мира объектов»: определение основных объектов задачи, их наилучшее представление на экране, выбор операций преобразования этих объектов.

Данная работа посвящена организации диалога персональной СУБД [2] реляционного типа с использованием алфавитно-цифрового дисплея. СУБД реализована в ОС RT-11 (РАФОС) в системе программирования Паскаль и может выполняться на ЭВМ СМ1300, СМ4, СМ1420, ДВК2, ДВК3 и аналогичных.

Три взгляда на реляционную БД

Одна из причин большого числа реализаций и популярности именно реляционных СУБД на мини- и микроЭВМ — простота реляционной модели данных. В ней данные — это набор «таблиц»-отношений с поименованными «столбцами»-атрибутами. В каждом из «столбцов» могут храниться данные только одного типа (поэтому и домен можно ассоциировать с типом данного: «целые числа», «текстовые строки» и т. д.).

Очевидно, что объект, соответствующий отношению, должен представлять «электронной таблицей», часть ко-

торой видна на экране. Принципиальных ограничений на число колонок и строк в таблице не налагается, следовательно, в таблице можно передвигать окно-экран в вертикальном и горизонтальном направлениях, указывая текущее положение в ней. Часть таблицы, видимую на экране, можно редактировать, как в экранном текстовом редакторе. Движение по отношению и его редактирование — это навигационные операции в терминах [3].

Такое общее представление одного отношения в СУБД с непосредственным редактированием, хотя и естественное для заполнения «электронной таблицы», все еще не адекватно возможностям реляционной СУБД.

Во-первых, в реляционной СУБД необходимо выполнять спецификационные операции [3], результат которых — новые отношения. Это теоретико-множественные (объединение, пересечение, разность, симметрическая разность) и реляционные (выборка, проектирование, соединение) операции. Поэтому при манипуляциях с отношениями необходимо изображать на экране до трех отношений (один или два аргумента и результат).

Во-вторых, к естественным манипуляциям с отношением как с таблицей относятся создание отношения, увеличение-уменьшение ширины отдельных колонок на экране, перестановка-удаление-переименование отдельных колонок. Эти операции, очевидно, должны выполняться над отношением, представленным в форме, отличной от формы «электронной таблицы».

И, наконец, в-третьих, нужно как-то решить проблему отображения текстовых строк, по длине превышающих ширину колонки на экране дисплея. Если при визуализации отношения длинное текстовое значение атрибута можно разместить в нескольких строках с переносами, то при экранном редактировании такое представление крайне нежелательно.

При проектировании диалога нужно стремиться к интуитивной согласованности объекта и «обрабатывающих» операций, предоставляя человеку максимальную свободу действий в рамках текущего «мира». Если придерживаться этого правила, то возможно наличие в СУБД функций: 1) создания-удаления-обслуживания одного отношения и выполнение спецификационных операций; 2) редактирования отношения; 3) визуализации отношения определяет не один, а целых три «Мира отношений».

Первый — «Мир доступа», в котором

пользователь может работать одновременно с тремя отношениями, либо используя их как аргументы в спецификационных операциях, либо преобразуя каждое из них с «точностью до столбца».

Для экранного редактирования содержимого каждого из трех отношений предусмотрен «Мир редактирования», в который можно войти из «Мира доступа», указав на нужное отношение и нажав клавишу ВОЙТИ. Клавиша ВЫЙТИ возвращает из «Мира редактирования» в «Мир доступа».

«Мир просмотра» спроектирован для визуализации на экране содержимого отношения через подвижное окно-экран.

Диалог при работе в этих «мирах» в соответствии с принципом непосредственности состоит в перемещении курсора по определенным полям на экране, редактировании этих полей вводом текста и нажатием на управляющие клавиши (рис. 1). Действие каждой из них зависит от «мира» и контекста, в котором осуществляется нажатие. Это приводит к огромному разнообразию результатов, получаемых вследствие манипулирования ограниченным количеством клавиш. В то же время контекст всегда подсказывает результат выполнения операции. Тип диалога, при котором всегда «параметры» вводятся перед «командами», не ограничивает пользователя рамками определенного сценария, потому что порядок ввода параметров при непосредственном редактировании несуществен.

Клавиша ПРЕФИКС используется для получения новых или модификации старых команд. Новые команды образуются так: (П) + (первая буква команды), например, ВЫБОРКА (П «В»).

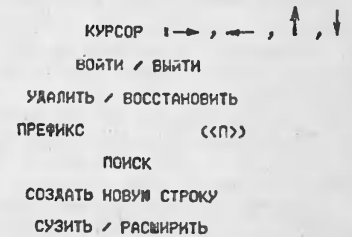


Рис. 1. Основные клавиши управления

Пример работы с СУБД

На рис. 2 показано некоторое состояние «Мира доступа». На экране видны три отношения. Первые два из них — существующие с именами FUNCT и ORDERS, третье — предполагаемый результат выполнения спецификационной операции.

Под строкой с именем отношения, количеством строк и датами создания и модификации в выделенной «полосе» изображаются атрибуты. Такое изображение отношений можно понимать как отдельно взятые заголовки и шапки таблиц, содержимое которых погружено в «Мир редактирования».

ОТНОШЕНИЕ1>DX1:FUNCT					СТРОК: 342					СОЗДАНО: 01.02.88					МОДИФ.: 12.03.88						
П2		01			П4		02			П5		03									
		АРГУМЕНТ					ФУНКЦИЯ 1					ФУНКЦИЯ 2									
У /Ф >N->1 SIN(U(1))/V(1)*#2->2 V(2)+R(1)->1 SQRT(U(1))->3 U(3)->0																					
ОТНОШЕНИЕ2>ORDERS					СТРОК: 85					СОЗДАНО: 12.11.87					МОДИФ.: 30.03.88						
Д		04		+1		Ц		06		+2		С		073		?		08			
		ДАТА				N		ЗАКАЗА				АДРЕС				????????????????				
У /Ф >																					
ОТНОШЕНИЕ3>RESULT					СТРОК:					СОЗДАНО:					МОДИФ.:						
У /Ф >																					

Рис. 2. Примерное состояние экрана в «Мире доступа»

ОТНОШЕНИЕ >PHONES					СТРОК: 332					СОЗДАНО: 23.12.87					МОДИФ.: 31.03.88							
#		С		01			С		02			С		03								
П/П				ФАМИЛИЯ					ТЕЛЕФОН					АДРЕС							
330				ПЕТРЯКОВ СЕМЕН ГРИГОРЬЕВИЧ			75-06-33					Г.Львов, Ул.Кузн										
331				КАШКИН ЕВГЕНИЙ ИВАНОВИЧ			33-26-08															
332				МАЛАХОВА ЕЛЕНА ПЕТРОВНА			75-34-23					КРЬНОК, ДОМ. N 7, >										
ПОИСК><(N)0&(U(1)*#КОР'>!(U(1)=E)																						

Рис. 3. Примерное состояние экрана в «Мире редактирования»

По изображению колонок можно определить количество позиций, отведенных для каждой из них на экране, тип значений (Ц — целые, П — числа с плавающей запятой, действительные, Д — дата, В — время, С — текстовые строки), для действительных чисел — количество цифр после запятой (точность) при выводе на экран. Кроме этого, отображается номер каждой колонки, информация об отсортированных и защищенных от записи атрибутах.

С помощью курсора пользователь может передвигаться по всему экрану и редактировать представление отношений или указывать на отдельное отношение как на аргумент, например для выполнения выборки.

Визуальные характеристики второго отношения таковы, что целиком его атрибуты не помещаются на экран. В этом случае можно сдвигать их изображения вправо-влево. В примере первые три атрибута ORDERS «находятся» слева от экрана, а 9, 10 и т. д. — справа. Сдвиг всей «шапки» происходит в нужном направлении при попытке пересечь курсором границу экрана для того, чтоб «попасть» в нужный атрибут.

По изображению второго отношения видно, что оно отсортировано по датам, а в пределах одинаковых дат — по номерам заказов. Пятая колонка в этом отношении «невидимая», т. е.

имеет нулевую ширину при выводе на экран. Значения седьмой колонки защищены от записи (признак «З»). Восьмая колонка создается, ее имя и тип еще не определены.

В первом отношении пользователь предполагает провести вычисления. Для этого в экранном режиме в строке (У/Ф) УСЛОВИЕ/ФУНКЦИЯ записано несколько выражений. В вычислениях значения колонок участвуют как векторы. Смысл выражений для первого отношения на рис. 2 таков: 1) пронумеровать первую колонку; 2) во вторую колонку записать значения функции $\sin(x)/x^2$, где x — значения из первой колонки; 3) в первом регистре (с каждым отношением связано 8 регистров, аналогичных регистрам калькулятора) накопить сумму всех значений второй колонки; 4) в третью колонку записать значения функции \sqrt{x} , где x — значения

из первой колонки; 5) значения из третьей колонки вывести на экран дисплея.

При редактировании отношения (телефонного справочника) содержимое его модифицируется в экранном режиме. При попытке перейти в столбец, который не уместился на экране, «экран передвигается» в нужную сторону. Значения атрибута АДРЕС могут превышать ширину соответствующей колонки. В этом случае на экране видно только начало адреса. Если необходимо ввести много символов или просмотреть весь адрес, то в соответствующей строке текстового атрибута с помощью клавиши ВОЙТИ организуется индивидуальное окно, своеобразный «Мир строки». На рис. 3 окно обозначено <...>. Через него можно просмотреть всю строку или скорректировать ее в экранном режиме. В строке ПОИСК находится логическое выражение, используемое для останова движения по отношению, индицированному клавишей ПОИСК. В данном случае условие окончания поиска: либо после десятой найдена строка с фамилией, заканчивающейся на КОР, либо найдена строка, в которой нет фамилии.

«Мир доступа»

Первоначально на экране изображены три пустых заголовка (или шаблона) для отношений. Каждый шаблон (рис. 4) — это, фактически, заголовок и шапка таблицы. Шаблоны предназначены для вызова существующих или создания новых отношений. Указанные на рис. 4 поля могут быть открыты или закрыты для ввода. Только информационными являются поля 3, 4, 5 и 13, предназначенные для вывода количества строк в отношении, даты создания-модификации и сообщения об ошибке. К этому же типу относятся поля индикации защиты атрибута от записи (10) и сортировки (11).

В начале работы открыты только поля 2 — для имени отношения. Ввод имени отношения в это поле вызывает либо заполнение заголовка и шапки информацией об атрибутах (в случае, если такое отношение найдено), либо начало процесса создания нового отношения. В последнем случае для третьего шаблона новое отношение не создается, а введенное имя определяет имя отношения, которое будет получено впоследствии при выполнении спецификационной операции.

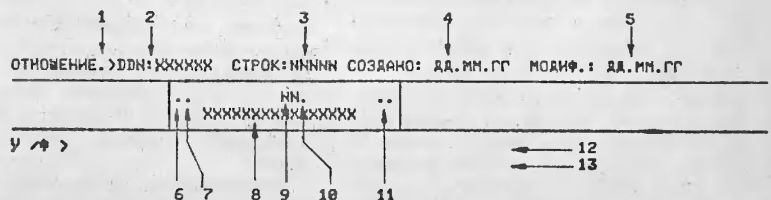


Рис. 4. Основные поля в «Мире доступа»

Создать новое отношение — это значит определить его имя и атрибуты. Новая колонка создается командой АТРИБУТ (⟨П⟩ «А»). В создаваемом атрибуте нужно ввести его имя поверх 16 знаков вопроса (см. рис. 2, колонку 8; рис. 4, поле 8) и тип значений (Ц, П, Д, В или С в поле 6 рис. 4). Если введен тип хранения — действительные числа, то в поле 7 можно ввести точность их представления при выводе на экран. Новые колонки можно создавать и в существующих отношениях.

Место, в которое вызвано отношение, играет определенную роль. Отношения 1 и 2 — это первый и второй аргументы во всех двуместных спецификационных операциях. Отношение 3 — результирующее.

Изложенные правила позволяют вызывать на экран старые отношения, создавать новые или определять результирующее в любом порядке. При создании отношения можно модифицировать практически все поля — имя отношения, имена атрибутов и их типы. В существующем отношении можно менять только имена атрибутов. В любом случае возможны операции над атрибутами: расширять-сужать, переставлять, удалять-восстанавливать. Атрибуты меняются местами при вводе в поле номера атрибута (9 на рис. 4) нового номера. Например, если в поле номера *i*-го атрибута введено число *j*, то *i*-й и *j*-й атрибуты меняются местами. Вводом нового номера отношения в поле 1 шаблона можно переставлять на экране сами отношения.

Каждое нажатие клавиши СУЗИТЬ уменьшает на одну позицию ширину указываемого курсором атрибута. Если нажать ее достаточное число раз, атрибут станет «невидимым» на экране. Клавиша РАСШИРИТЬ имеет обратное действие. Для «материализации» невидимых атрибутов применяется правило: если при расширении атрибута у него есть соседние невидимые, то они становятся видимыми шириной в одну позицию. После этого можно установить курсор в эту единственную позицию и определить желаемую ширину. Поэтому последний видимый атрибут нельзя сделать невидимым.

Если курсор находится в поле имени отношения и нажимается клавиша УДАЛИТЬ, то все изменения атрибутов становятся недействительными, поля заголовка и шапки очищаются, превращаясь опять в пустой шаблон. Для окончания работы с отношением с сохранением всех изменений используется клавиша ВЫЙТИ. При определении нового отношения нажатие клавиши ВЫЙТИ приводит к его фактическому созданию.

Для вычислений отношений используется аппарат выражений. Выражения вводятся в экранном режиме в поле 12 (строку условий и функций). Выражения выполняются только после нажатия клавиши ФУНКЦИЯ (⟨П⟩ «Ф»).

Синтаксис выражения совпадает с

синтаксисом выражений Паскаля. Кроме обычных арифметических операций и отношений, имеется дополнительная операция пересылки «→» с приоритетом «+» или «-».

Оспраиваемыми могут быть числа, функции и текстовые строки. Если текстовая строка участвует в сравнении на равенство, то символ «*» имеет специальное значение: любая строка символов, в том числе пустая, а символ «%» — значение «любой символ». Если в операции сравнения участвуют текстовые строки разной длины, они выравниваются пробелами справа.

Функции, новые по сравнению с функциями Паскаля, которые можно использовать в выражениях, — это функции «значение *i*-го атрибута» ($V(I)$), «пустое значение» (E), «номер текущей строки» (N). При вычислении выражений действует такое правило: если было обращение к отношению, то выражение вычислять для всех строк, начиная с первой. Благодаря этому правилу можно трактовать функцию $V(I)$ как векторную и нет необходимости указывать в ней номер строки. Кроме выражения допускается использование условного оператора.

Семантика выполнения операции $F \rightarrow J$ такова. J — всегда целое число. Если $J > 0$, то F заносится в текущую строку в J -й атрибут. Если $J = 0$, то F выводится на экран в поле 13. Если $J < 0$, то F заносится в J -й регистр, связанный с отношением. Результатом выполнения операции $F \rightarrow J$ считается F .

Теоретико-множественные операции — это ⟨П⟩ «+» («объединение»), ⟨П⟩ «-» («разность»), ⟨П⟩ «*» («пересечение») и ⟨П⟩ «/» («симметрическая разность»). Перед выполнением операции нужно определить аргументы и имя результирующего отношения. Операция объединения требует, чтобы первые два отношения полностью совпадали по структуре, т. е. имели одинаковое количество атрибутов, имена и типы которых попарно должны совпадать (порядок несуществен). При выполнении операций «разность» и «симметрическая разность» несовпадение структуры атрибутов приводит к тому, что все сравнения выполняются только по совпадающим атрибутам, а в результирующее отношение выводятся подходящие строки первого отношения. Операция ⟨П⟩ «*» для отношений с несовпадающими полностью атрибутами выполняется как естественное соединение [4].

Операция выборки выполняется после нажатия клавиши ⟨П⟩ «В». В строке У/Ф нужно задать логическое выражение «фильтр». В результирующее отношение переписываются строки, удовлетворяющие записанному условию. Если «фильтр» не указан, то выборка эквивалентна простому копированию или проекции (если из исходного отношения удалены некоторые колонки).

Возможность переставлять отношения на экране позволяет результаты делать аргументами дальнейших опе-

раций. Следовательно, можно вычислить как угодно сложное реляционное выражение.

Порядок сортировки отношения определяется указанием очередного ключа (сортируемого атрибута) и вводом команды ⟨П⟩ «↑» (сортировка по возрастанию) или ⟨П⟩ «↓» (сортировка по убыванию). Сортировка выполняется после нажатия клавиши ВЫЙТИ.

Команда ЗАЩИТА (⟨П⟩ «3»), введенная в любом из полей атрибута (6—9), устанавливает защиту этого атрибута от записи. Та же команда, введенная в поле имени отношения при создании нового отношения, служит для присваивания пароля, по которому в дальнейшем разрешается доступ к секретному отношению.

Для распечатки содержимого отношения с переносами длинных текстовых строк и расположением материала по листам предназначена команда ТВЕРДАЯ КОПИЯ — ⟨П⟩ «Т».

«Мир редактирования»

При нажатии клавиши ВОЙТИ происходит переход в «Мир редактирования» конкретного отношения. На экране отображается картинка (рис. 3). Находящиеся на экране 16 строк отношения можно редактировать, при необходимости смещая экран влево-вправо для доступа к не поместившимся на нем колонкам. Экран смещается при попытке пересечь его границу справа или слева. Организация с помощью клавиши ВОЙТИ на экране локальное окно, через него можно ввести или просмотреть значение текстового атрибута. Окно «закрывается» и пользователь продолжает движение по остальной части экрана после нажатия клавиши ВЫЙТИ.

Движение по отношению вверх-вниз осуществляется или с помощью «листаания» страниц клавишами ⟨П⟩ «↑»/⟨П⟩ «↓» (16 строк вверх-вниз), или при поиске. Поиск — это движение по отношению в указанном направлении, для очередной строки оценивается выражение в строке ПОИСК. Движение прекращается при достижении начала или конца отношения либо на строке, для которой условие принимает истинное значение.

В строке ПОИСК можно записывать и выражения, аналогичные выражениям «Мир доступа». При нажатии клавиши ФУНКЦИЯ они выполняются только для строки с курсором. Таким образом, можно организовать серийные вычисления или «экранный калькулятор».

«Мир просмотра»

Для просмотра всего отношения или его части предназначен «Мир просмотра». Чтоб войти в него, необходимо указать на интересующее отношение и ввести команду ПРОСМОТР.

Если в строке У/Ф было введено логическое выражение, то можно будет видеть только те строки, которые про-

ходят через этот «фильтр». Информация выводится в формате, близком к формату редактирования. Различие состоит в том, что, во-первых, попасть можно только в строку ПОИСК или в поле имени отношения, во вторых, значения текстовых атрибутов выводятся полностью, при необходимости — в нескольких строках экрана.

Если курсор находится в поле имени отношения, то клавиши КУРСОР, ВПРАВО/ВЛЕВО вызывают горизонтальный сдвиг экрана, открывая не уместившиеся на экране атрибуты, а клавиши ↑/↓ — вертикальное перемещение по отношению.

После вывода очередной страницы на экран можно изменить условие-фильтр и выводить строки по новому критерию. Если условие отсутствует, происходит просмотр всей таблицы. Выход в «Мир доступа» — после нажатия клавиши ВЫЙТИ.

Заключение

При проектировании описанного диалога авторы стремились реализовать

«чистое» непосредственное редактирование. Как и отмечено в [1], после выработки моторных навыков, на что уходит от нескольких минут до получаса, пользователь вообще перестает замечать систему как таковую и свободно действует в мире отношений.

Аппарат вычислений с помощью выражений и условных операторов, представление отношений «электронными таблицами» превращают описанную реляционную СУБД в интегрированную систему для обработки табличной информации. Данная система имеет элементы, родственные как электронным таблицам («Суперкалк» и др.), так и средствам работы с экранными формами СУБД (dBase и др.). Основное ее отличие от этих систем в том, что она предлагает единую модель обработки информации в рамках описанного диалога.

Представление «миров» может быть и другим, отличным от приведенного в статье. Однако диалог при доступе к реляционной базе данных будет характеризоваться переходами в трех «мирах» (доступ, редактирование, про-

смотр) с описанной иерархией подчинения.
290005, Львов, ул. Драгоманова, 14/16,
Львовское отделение Ин-та бисими
им. А. В. Паладина АН УССР;
тел. 72-85-08

ЛИТЕРАТУРА

1. Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации // Микропроцессорные средства и системы. — 1986. — № 1. — С. 44—46.
2. Корытный И. М. Информационная система в автоматизации экспериментальных исследований инструментальной и реализации // УСИМ. — 1987. — № 2. — С. 102—106.
3. Цикритзис Д., Лоховски Ф. Модели данных. — М.: Финансы и статистика, 1985. — 344 с.
4. Мейер Д. Теория реляционных баз данных. — М.: Мир, 1987. — 608 с.

Статья поступила 21.04.88

УДК 630.624.274

В. М. Гуненко, А. И. Антипов

КОМПЛЕКС ЗАДАЧ «АКТУАЛИЗАЦИЯ И ПРОГНОЗ ДИНАМИКИ ЛЕСНОГО ФОНДА»

В Отраслевой научно-исследовательской лаборатории по АСУ при Московском Лесотехническом институте разработан комплекс задач «Актуализация и прогноз динамики лесного фонда» (КЗ АПДЛФ) для проверки с помощью ЭВМ лесоводственных последствий принимаемых решений по объемам отпуска леса для главного пользования на уровне лесхоза, леспромхоза, лесовозной дороги. Он позволяет получить таблицы классов возраста и наиболее употребительные расчетные лесосеки на любую дату в пределах оборота рубки в зависимости от объемов главного пользования, задаваемых в диалоговом режиме, в разрезе хозсекций и последовательных десятилетних расчетного периода; определяются также неистощительные лесосеки.

Технически комплекс задач реализован в виде диалоговой системы на языке программирования БЕЙСИК для ПЭВМ «Искра 226» и рассчитан на пользователя, не имеющего специальной подготовки в области вычислительной техники. Время расчета одного объекта (с учетом ввода исходных данных) составляет полтора-три часа. При отсутствии ЭВМ для производства работ машинное время арендуется в других организациях из расчета 4 руб. за 1 час. В настоящее время проводятся работы по переводу программ комплекса на IBM, подобные ЭВМ.

КЗ АПДЛФ передан объединениям «Кареллеспром» и «Пермлеспром». Лаборатория может адаптировать комплекс к условиям других пользователей за три-шесть месяцев. Стоимость внесения изменений в алгоритмы и программы и обучение пользователей с передачей технической документации на гибком магнитном диске не превышает 5 тыс. руб.

Работа КЗ АПДЛФ может быть продемонстрирована в НАТИ или у заказчика в случае оплаты командировочных расходов на двух человек.

141001, Мытищи Московской области, МЛТИ, ОИИЛ по АСУ, Антипов А. И.; тел. 582-47-50

Сообщение поступило 1.11.88

УДК 681.2.06

В. Н. Авдеев, А. Е. Белов, А. Ю. Жедь, В. Ф. Корнюшко

ИНТЕРПРЕТАТОР ЯЗЫКА БЕЙСИК СО ВСТРОЕННЫМ ГРАФИЧЕСКИМ ПАКЕТОМ ДЛЯ ДВК

Основой данной разработки является широко используемая версия интерпретатора БЕЙСИК ОС РТ-11. Встроенный графический пакет поддерживает черно-белую графику на ДВК3 и цветную графику на моделях ДВК, снабженных адаптером ЦДР и цветным монитором.

Подпрограммы пакета написаны на языке ассемблера, что обеспечило высокую скорость выполнения графических построений при малом объеме выполняемого кода. Пакет прикомпоновывается к языку интерпретатора при генерации. Исполняемый объем интерпретатора занимает около 16 Кбайт в ОЗУ (в неоверлейном варианте).

Пакет включает набор примитивов, ориентированных на различные математические приложения. Все операторы пакета унифицированы и не зависят от аппаратной части комплекса.

Исключение медленных процессов компиляции и компоновки и отказ от непомерных запросов на дисковое пространство, автоматическое преобразование типов параметров в вызовах — все это выгодно отличает данный программный продукт от многочисленных версий графических библиотек для Фортрана и Паскаля.

КРАТКИЕ СООБЩЕНИЯ

Форма вызова	Время выполнения, с	Описание
PNT (X, Y, ЦВ)	—	Вывод точки указанного цвета с абсолютными координатами X и Y
ASG (X, Y [, ЦВ])	0,011	Создание окна с границами X1 и X2, Y1 и Y2
SCALE ([N,] X1, X2, Y1, Y2)	0,014	Масштабирование экрана или окна
POINT ([N,] X, Y, ЦВ)	0,013	Вывод точки указанного цвета в масштабных координатах окна
PLOT ([N,] X1, X2, Y1, Y2, ЦВ)	—	Построение прямой указанного цвета в окно между точками (X1, X2) и (Y1, Y2)
CIR ([N,] X, Y, РАД, ЦВ)	0,08	Построение окружности
SYM ([ЦВФ,] X, Y, КОД [ЦВ, РАЗМ])	—	Вывод символа
CLR ([N,], ЦВ) CLR ([N,], X1, X2, Y1, Y2, ЦВ)	—	Очистка экрана или окна заданным цветом
MASK (N)	0,013	Установка или снятие маски чтения
STAT (N)	—	Установка режима работы видеотерминала
HC	—	Получение твердой копии изображения

Основной набор графических примитивов (см. таблицу) позволяет работать с цветом, проводить линии, масштабировать координатную сетку, работать с отдельными областями, выводить алфавитно-цифровую информацию. Кроме того, имеются операторы для получения твердой копии изображения на устройстве печати, программно совместимом с принтером EPSON, и записи-чтения изображения на (с) диск (диска). Экранный редактор строк позволяет отказать от оператора SUB и значительно ускорить отладку программ.

Интерпретатор с пакетом включен в учебную меню-систему и применяется в лабораторных практикумах по курсам «Вычислительная математика и ЭВМ» и «Применение микроЭВМ в отрасли». Графический пакет использован при построении программных тренажеров для организации технологических игр и отработки у химиков-технологов навыков управления технологическими процессами, организации вычислительных экспериментов в задачах планирования и обработки экспериментов, а также при создании автоматизированных контролируемых и обучающих систем.

119831, ГСП, Москва, М. Пироговская, 1,
МИТХТ; тел. 246-46-50

Сообщение поступило 13.04.88

Обзор программных продуктов системы «Интерфейс внешних функций интерпретатора Фокал БК-0010»

Если Вы имеете микрокомпьютер «Электроника БК-0010» с интерпретатором языка Фокал, то интерфейс внешних функций (ИВФ) позволит повысить эффективность его использования. Добавляя к интерпретатору функции в кодах, которые будут работать как новые операторы или процедуры, Вы расширите язык и сможете оптимизировать Ваши программы. Функции, будучи подпрограммами в кодах, работают в десятки раз быстрее, чем аналогичные программы на Фокале, и занимают гораздо меньше памяти. В то же время внешние функции включаются в интерпретатор, как «родные», и все преимущества языка высокого уровня сохраняются, ИВФ защищает от программ на Фокале 8 Кбайт ОЗУ, резервируя память для внешних функций данных. Данные в этой области могут храниться в любом формате. Это экономит память, ведь интерпретатор расходует ее для данных очень расточительно. Кроме того, обращение к данным в область ИВФ будет осуществляться за фиксированное время, независимо от объема данных, а не так, как в собственной области программ на Фокале. Это дает экономию времени исполнения программ и памяти, ведь в области ИВФ числа можно хранить байтами и словами по два байта, а не по восемь, как это делает интерпретатор.

А вот другая проблема использования Вашего компьютера. Если Вы написали прикладную программу год назад, не включили в нее средства диалога и она долго не использовалась, то Вам, наверное, трудно теперь вспомнить, как с ней работать. Тем более непросто все растолковать другому пользователю, особенно если он не программист. А когда таких пользователей вокруг Вас много, то лучше было бы файлы с этой программой стереть или потерять!

Для преодоления этой проблемы Вы можете воспользоваться системой внешних функций «музыка-цвет-джойстик». Занимая немного памяти, Вы будете значительно повышать эффективность использования Ваших программ. Возможна реализация «выдвижных» меню, «приборных» табло, оконного взаимодействия, причем общение с компьютером через клавиатуру можно полностью или почти полностью исключить. Так организуются интерфейсы с пользователем по принципу «что вижу, то и имею». Музыкальные сигналы помогут ориентироваться в состояниях программы — получится интерфейс: «что вижу и слышу, то и имею». Наибольший эффект будет достигаться при использовании цветного монитора и указателя ин-

СОВМЕСТНОЕ СОВЕТСКО-АМЕРИКАНСКОЕ ПРЕДПРИЯТИЕ «ДИАЛОГ»

Фирма № 1 по разработке программного обеспечения — корпорация MICROSOFT (R) предлагает свои продукты.

Единственный официальный представитель фирмы на территории СССР — Совместное советско-американское предприятие «ДИАЛОГ».

Купив у СП «ДИАЛОГ» оригинальные программные продукты MICROSOFT (R), Вы станете владельцем фирменного пакета с полным комплектом документации, всех необходимых приложений и справочных материалов.

Став зарегистрированным пользователем фирмы MICROSOFT (R), Вы приобретаете:

- доступ к системе поддержки пользователей («горячая линия») с возможностью выхода на «hot line» фирмы MICROSOFT (R);
- право на информационное обслуживание (информация о новых версиях и программных продуктах MICROSOFT (R));
- гарантию отсутствия вирусов на дискетах;
- возможность покупки новых версий ПП по сниженным ценам.

СП «ДИАЛОГ» обязуется предоставлять исчерпывающую информацию о всех семинарах и выставках, проводимых фирмой MICROSOFT (R) и СП «ДИАЛОГ».

Стратегия и тактика совместного советско-американского предприятия «ДИАЛОГ» — обеспечение гарантий наших партнеров!



Если Вас заинтересуют наши предложения, обращайтесь по телефону: 329-45-33

Адрес: 115598, г. Москва, ул. Ягодная, 17. СП «ДИАЛОГ»

Проезд: м. Ленино, авт. 701, 203 до ост. Загорьевская, 10

Почтовый адрес: 107066, Спартаковская, 13. СП «ДИАЛОГ»

MICROSOFT В OPTIMIZING COMPILER VERSION 6,0 ОПТИМИЗИРУЮЩИЙ КОМПИЛЯТОР ДЛЯ ЯЗЫКА СИ ФИРМЫ MICROSOFT. ВЕРСИЯ 6.0

Что нового позволяет делать Оптимизирующий транслятор для языка Си фирмы Microsoft? Полная поддержка MS-OS/2 позволяет Вам перешагнуть через 640-килобайтный барьер. Динамическое редактирование связей и соединение программ, поддержка программ, работающих как под OS/2, так и под MS-DOS, редактор связей и улучшенный текстовый редактор для привилегированного режима. Оптимизация за счет как непосредственной подстановки кода, так и вычисления констант при компиляции, подключение Microsoft QuickC для быстрой компиляции и макетирования программ, способность отлаживать очень большие программы с помощью отладчика CodeView — все это дает Вам, профессиональным программистам, то, без чего нельзя жить: скорость!

Быстрота выполнения программ

Оптимизирующий компилятор фирмы Microsoft версии 6.0 создает самый быстрый для персональных компьютеров код с языка Си. Компилятор преобразует несколько десятков функций в непосредственный код. Он также включает оптимизацию за счет удаления инвариантных блоков из циклов и за счет автоматического размещения на регистрах переменных, используемых внутри циклов.

Оптимизирующий компилятор фирмы Microsoft также помогает оптимизировать Ваш код во время его написания. Техника кодирования для получения наиболее быстрых программ включена в документацию. Вы можете смешивать модели памяти, используя ключевые слова NEAR, FAR и HUGE. Вы можете объявить

регистровые переменные для информации компилятору. Вы можете использовать соглашения Паскаля о передаче параметров для того, чтобы сделать программы меньшими и более быстрыми и для возможности их использования с пакетом Microsoft Windows.

Быстрая компиляция и макетирование с помощью QuickC

Вы можете компилировать Ваши программы со скоростью 25.000 строк в минуту с помощью компилятора QuickC, включенного в пакет Microsoft Си. Благодаря тому, что он запускается только с помощью нескольких нажатий клавиш, Вы можете использовать QuickC во время разработки для наиболее быстрой компиляции, а затем снова использовать Оптимизирующий компилятор для достижения наибольшей скорости выполнения.

QuickC служит идеальным инструментом для макетирования. Все программные средства, которые Вам нужно для этого — редактор, программа MAKE, компилятор и отладчик — объединены в одной программной среде.

Редактор позволяет писать исходный текст легко и эффективно. Вы можете вырезать, копировать и склеивать все — от отдельных символов до всей программы. Перед компиляцией компилятор автоматически создает MAKE-файл с информацией, требуемой для построения программы. Он хранит информацию о 26 синтаксических ошибках, возникших при компиляции — поэтому Вам не требуется перекомпилировать программу для обнаружения каждой новой ошибки. Редактор автоматически устанавливает курсор на каждую из ошибок. Отладчик позволяет устанавливать точки останова, пошагово выполнять программу, постоянно отображать значения переменных. После этого требуется обработать программу с помощью Оптимизирующего компилятора версии 6.0 для получения наиболее быстрой программы.

Быстрая отладка с помощью отладчика фирмы Microsoft codeView

Microsoft CodeView — это прекрасный отладчик, ориентированный на работу с окнами, позволяющий отлаживать на уровне исходного текста. Его наглядность, настроенные на языки высокого уровня команды и непосредственная помощь позволяют легко отлаживать программы. Многочисленные окна дают полный контроль над всем ходом выполнения программы в ходе отладки.

Под MS-OS/2 отладчик CodeView позволяет отлаживать программы объемом до 128 МВ. Имеется поддержка расширенной памяти (EMS), допускается отладка оверлейных программ, написанных на нескольких языках фирмы Microsoft, включая QuickBasic, Basic-компилятор, Си, QuickC, Фортран, Макроассемблер и Паскаль — все в одном отладчике.

Оптимизация кода для выигрыша в скорости

Оптимизирующий компилятор фирмы Microsoft — самый быстрый компилятор, существующий на персональных компьютерах. Оптимизация включает хранение наиболее часто используемых выражений в регистрах, индукцию переменных, переводящую умножения в сложение, подстановку констант, которые могут быть вычислены во время компиляции. Также в несколько раз ускорен ряд библиотечных функций, таких как atan, cos, sqrt.



Быстрое макетирование с помощью QuickC

Microsoft Си версии 6.0 включает QuickC версии 2.0, который позволяет редактировать, компилировать, отлаживать и выполнять код в одной инструментальной системе.

Встроенный редактор

Редактор QuickC помогает писать исходный текст быстро и легко с функциями, включающими выбор режима вставки/замены, поддержки 43-строчного режима (на видеоадаптерах EGA), поиск парных скобок, соединение различных блоков текста. Также имеется клавиша, позволяющая переключаться между двумя файлами для быстрого просмотра.

Автоматическое создание MAKE-файлов для построения программ

Microsoft QuickC дает Вам эффективный и продуманный путь для сборки Вашей программы. Вам требуется просто указать модули, из которых состоит программа, и Microsoft QuickC автоматически создаст MAKE-файл для построения Вашей программы. Затем служебная программа MAKE перекомпилирует и соберет Вашу программу только из тех модулей, которые изменились.

Быстрая компиляция

Microsoft QuickC компилирует Си-программы со скоростью 25 000 строк в минуту. Диалоговый интерфейс позволяет легко выбрать ключи для компиляции, такие как уровень выдачи предупреждений или направление компиляции (диск/память).

Встроенный отладчик

Отладчик Microsoft QuickC, являющийся подмножеством отладчика Microsoft CodeView, позволяет пройти по Вашей программе, задать переменные для просмотра, установить точки останова и, используя стек возвратов, узнать какие вызваны функции.

Полная совместимость с Microsoft Си 6.0

Microsoft QuickC версии 2.0 полностью совместим как по текстам, так и по объектному коду с Оптимизирующим компилятором фирмы Microsoft версии 6.0. Он создает MAKE-файл в стандартном формате, используемом Си версии 6.0 для построения программ, отлаживаемых с помощью CodeView. Также используются одни и те же ключи для компиляции и редактирования связей, что обеспечивает быструю компиляцию с помощью QuickC при вызове его с помощью командных строк.

Возможности отладки при использовании Microsoft CodeView

Отладчик CodeView, включаемый в пакет вместе с Оптимизирующим компилятором фирмы Microsoft версии 6.0, является окном-ориентированным, поддерживающим отладку на уровне исходного текста. Он обеспечивает полное управление выполнением программы под MS-DOS или MS-OS/2. С помощью MS-OS/2 Вы можете отлаживать большие программы (объемом до 128 MB), многопроцессные программы и динамически загружаемые модули.

Полный дружественный интерфейс

Многочисленные окна CodeView (Отображение, Диалог, Просмотр, Регистры) позволяют Вам одновременно видеть исходный текст, реассемблированный текст,

переменные, регистры и стек. В ходе отладки программы значения автоматически меняются на экране. Возможна установка условных точек останова и замедленное автоматическое пошаговое выполнение программы. В дополнение мощная система меню позволяет Вам не запоминать имена команд.

Отладка на уровне исходного текста

С помощью CodeView Вы можете отлаживать программы, используя исходный текст программы со всеми ее переменными. При этом можно допускать или не допускать отображение в ассемблерном виде.

Богатство ключей для настройки отладчика

Вы можете использовать выражения, допустимые в конкретных языках, и вызывать отдельные функции программы прямо с клавиатуры. Возможно изменение любой переменной по ее имени, включая глобальные, локальные и регистровые переменные. Вы также можете отображать структуры данных и следить за списками и вложенными структурами данных. Смена экранов позволяет отлаживать графические программы путем отображения текста и вывода программы на различных страницах экрана.

Наличие более 400 библиотечных подпрограмм

Оптимизирующий компилятор фирмы Microsoft версии 6.0 поставляется с более чем 400 библиотечными программами, включая функции для управления динамической областью, отладки работы с динамическими областями и функциями времени. Также имеется подпрограмма пептах, возвращающая количество свободной памяти для Вашей программы.

Компилятор Microsoft Си версии 6.0 включает все библиотеки ANSI Си; эти библиотеки совместимы с библиотеками, имеющимися для системы UNIX System V. Также Microsoft Си поддерживает около 70 различных библиотек.

Расширенная библиотека графических функций

Новая графическая библиотека позволяет создавать захватывающие программы, использующие все графические возможности экрана. Она включает дополнительные библиотечные подпрограммы, служащие для управления палитрой, атрибутами, вывода и преобразования образов.

MICROSOFT WINDOWS

Программа взаимодействия с пользователем

Microsoft Windows — это многозадачность в однозадачной операционной системе MS-DOS. Microsoft Windows — это удовольствие от работы с компьютером. Microsoft Windows — это единственная операционная среда, которая предоставит Вам наилучший путь к OS/2, операционной системе будущего.

MICROSOFT C OPTIMIZING COMPILER VERSION 6.0 ОПТИМИЗИРУЮЩИЙ КОМПИЛЯТОР ДЛЯ ЯЗЫКА СИ ФИРМЫ MICROSOFT. ВЕРСИЯ 6.0

Для персональных компьютеров, работающих под управлением операционной среды OS/2 или MS-DOS фирмы Microsoft. Быстрый код, быстрая компиляция, быстрое макетирование и быстрая отладка. Все это предоставляет Оптимизирующий компилятор для языка Си фирмы Microsoft — последнее слово в развитии языка Си.

Не удивительно, что на нем остановили свой выбор профессиональные программисты.

ПУЛЬТ ОТЛАДОЧНЫЙ «РАСТР»

Пульт предназначен для комплексной отладки разрабатываемых пользователем программ и их занесения в ППЗУ однокристалльного цифрового процессора обработки сигналов (ЦПОС) типа KM1813BE1, используемого в аналоговых подсистемах геофизического и других видах отечественного приборостроения.

Кроме того, пульт обеспечивает считывание программы из ППЗУ ЦПОС в ОЗУ пульта, сравнение ее с исходной программой и испытание ЦПОС на реальных или тестовых сигналах.

Пульт рассчитан на работу в каротажных лабораториях, отопляемых автобусах, специально оборудованных прицепах при температуре окружающей среды $+10 \dots +45^\circ\text{C}$ и относительной влажности 90% при $+30^\circ\text{C}$.

Работа пульта основана на выполнении встроенным микропроцессорным контроллером различных операций в соответствии с его программным обеспечением, реализованным на БИС ППЗУ.

Управление работой пульта осуществляется при помощи команд оператора, подаваемых с блока клавиатуры.

В качестве внешних устройств, расширяющих функциональные возмож-

ности пульта, могут быть использованы:

- бытовой кассетный магнитофон;
- генератор внешних синхросигналов для работы ЦПОС.

Пульт обеспечивает также работу по интерфейсу ИРПР с микроЭВМ или печатающим устройством.

Пульт выполнен в виде настольного прибора, состоящего из блока программирования и испытателя, и блока функциональной клавиатуры.

Технический уровень соответствует мировому и защищен авторскими свидетельствами СССР № 1339570 и № 24734. Отечественных аналогов не имеет.

Основные технические характеристики:

1. Режимы работы пульта: программирование, испытание, тест.
2. Пульт обеспечивает:
 - запись данных на кассетный бытовой магнитофон;
 - считывание данных с кассетного бытового магнитофона;
 - сопряжение с внешними устройствами по интерфейсу ИРПР.
3. Пульт реализует следующие функции:
 - ускоренный ввод программы на языке ассемблер ЦПОС как с блока клавиатуры, так и с кассетного магнитофона и ее отражение на экране встроенного дисплея;

— трансляцию программы с языка ассемблер ЦПОС в объектный код;

— отладку программ пользователя на программной модели ЦПОС с использованием точек останова и отображением на дисплее текущего состояния ЦПОС;

— занесение отлаженной программы в ППЗУ ЦПОС и сравнение ее с исходной программой;

— испытание запрограммированного ЦПОС на реальных или тестовых входных сигналах.

4. Питание пульта от сети напряжением 220 В, 50 Гц.

5. Потребляемая мощность, не более 90 ВА.

6. Габаритные размеры, мм: блока программирования и испытателя — $480 \times 305 \times 405$;

блока клавиатуры — $430 \times 150 \times 65$.

7. Масса без упаковки, кг: блока программирования и испытателя — 17;

блока клавиатуры — 1,5.

В комплект поставки входят:

- блок программирования и испытателя «РАСТР»;

— блок клавиатуры «РАСТР»;

— комплект запасных частей, инструмента и принадлежности (ЗИП);

— эксплуатационная документация.

Цена — 7250 руб.

Заявки на приобретение пульта направлять по адресу: 450001, Башкирская АССР, г. Уфа, ул. Комсомольская, 2.

КОМПАКТНЫЙ ЭМУЛЯТОР МИКРОПРОЦЕССОРА M1821VM85A ДЛЯ ЛЮБОЙ ЭВМ

Эмулятор EMU является однопроцессорным средством отладки на основе однокристалльного микропроцессора M1821VM85A и других микропроцессоров, являющихся аналогами 18085. Работает под управлением ПЭВМ (PC, PC/XT, PC/AT, EC1840, EC1841, Нейрон, Искра 1030), микро- или мини-ЭВМ и позволяет проектировать и вести отладку аппаратных и программных средств микропроцессорных систем в реальном масштабе времени.

Технические характеристики

Объем эмулируемой памяти — не менее 52 Кбайт

Зонирование памяти — через 4 Кбайт

Максимальная частота тактирования — 3072 кГц

Минимальная частота тактирования — 500 кГц

Управление эмулятором от внешней ЭВМ по последовательному каналу типа ИРПР или С2 со скоростью 9600 Бод.

Источники питания внешние:

+5 В (5%) не более 0,7 А; +12 В (10%) не более 0,08 А; —12 В (10%) не более 0,08 А.

Встроенная система команд обеспечивает:

- ассемблирование;
- зонирование памяти;
- вызов подпрограмм с передачей параметров;
- дампование области памяти;
- заполнение области памяти константой;
- передача управления тестируемой программе с остановом в двух заданных точках разрыва;
- вычисление суммы и разности двух шестнадцатиричных чисел;
- чтение и индикация байта из порта;
- дизассемблирование содержимого области памяти (листинг);
- пересылка содержимого области памяти;
- вывод байта в порт;
- подстановка байта в память;
- трассировка тестируемой программы;
- профилирование программы пользователя;
- подсчет контрольной суммы области памяти;
- индикация и изменение состояния элементов процессора;

— запись на диск ЭВМ бинарного файла;

— загрузка файла с диска ЭВМ в эмулятор.

Габаритные размеры $265 \times 155 \times 25$ мм; вес 600 г.

Конструктивно эмулятор выполнен на одной печатной плате размерами 150×200 мм в металлическом корпусе. Элементная база включает микросхемы серии M1821, K555, K155, KP580, K 170 (45 микросхем).

В состав комплекта входят: устройство эмуляции EMU85; кабель для подключения к устройству;

соединитель СНО53-8 для подключения источника питания;

соединитель СНО52-30 для подключения к ЭВМ;

дискета для ЭВМ с программой обслуживания эмулятора;

техническое описание эмулятора;

инструкция пользователя эмулятора.

Срок поставки — не более 3 месяцев со дня заказа.

Заявки на приобретение направлять по адресу: 200034, г. Таллин, бульвар Сыprusе 219—73, кооператив Гриф. Телефон 537-994.

формации джойстик. Система «музыка-цвет-джойстик», основанная на ИВФ, остается открытой. Ее можно расширять, подключая через ИВФ новые функции.

Система программ, написанных на Фокале, названная PLOT, — это оконная графика. PLOT может хорошо дополнить систему «музыка-цвет-джойстик».

Все программные продукты включают в себя руководства по использованию. Вы найдете в них всю необходимую информацию для создания новых функций и прикладных программ.

Ниже приведены цены (в рублях) на программные продукты системы «Интерфейс внешних функций интерпретатора Фокал БК-0010».

1. Интерфейс внешних функций интерпретатора Фокал БК-0010: магнитная кассета с системой, общее описание, руководство системного программиста, руководство по использованию ИВФ — 280.
2. Система внешних функций «музыка-цвет-джойстик», версия 00: магнитная кассета с системой, руководство по использованию — 220.
3. Система внешних функций «музыка-цвет-джойстик», версия 01: магнитная кассета с системой, руководство по использованию — 280. Здесь необходим указатель информации джойстик. По особому заказу поставляются схема, эскиз конструкции и описание устройства.
4. Система программ машинной графики PLOT (Фокал): магнитная кассета с системой, руководство по использованию — 120.

Принимаются заказы на создание новых внешних функций. Цены ориентировочные. Устанавливаются на основе договоренности в зависимости от заказа на программные продукты и методическое обеспечение.

117049, Москва, 2-й Добрынинский пер., дом 5/9, кв. 39 («Контракт»); тел. 244-04-92 (вторн., четв. с 16-00 до 19-00).

Консультации: 142292, Московская обл., Пушкино, ИВФ АН СССР, А. П. Казанцев; тел. (Моск.): 923-74-67, доб. 2-93, 923-96-68 доб. 2-93

УДК 681.324

В. А. Богатырев

АРБИТР С ДЕЦЕНТРАЛИЗОВАННЫМ КОДОВЫМ УПРАВЛЕНИЕМ

В локальных вычислительных сетях для разрешения конфликтов обращения микроЭВМ к общей магистрали (ОМ) целесообразно использовать метод множественного доступа на основе децентрализованного кодового управления (ДКУ) [1], при котором все абоненты имеют различные приоритеты, задаваемые двоичными числами. Такого рода арбитр магистрали представлен на рис. 1. [2].

2 Микропроцессорные средства № 5

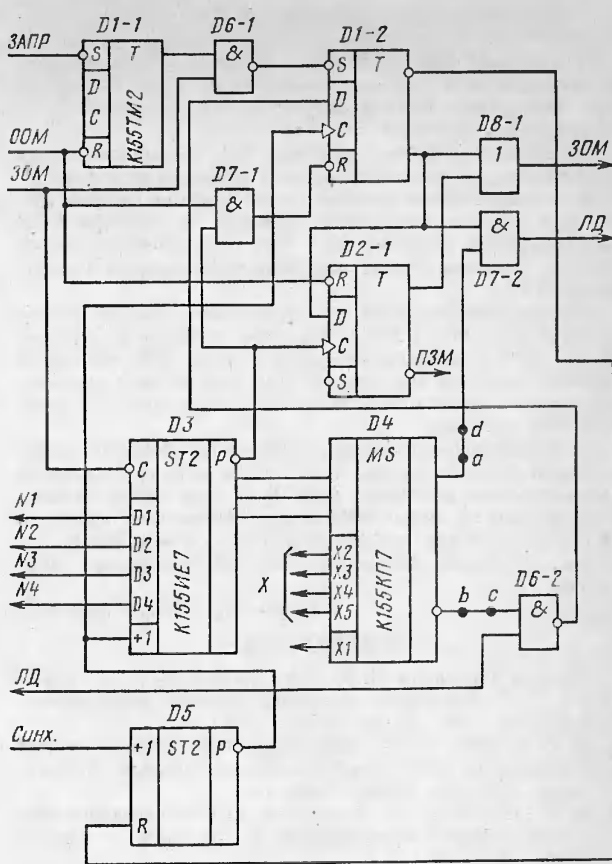


Рис. 1. Функциональная схема канала арбитра

Приоритет абонента задается двоичным кодом на входе X мультиплексора D4. Требующий захвата ОМ абонент, подавая сигнал на вход ЗАПР, заносит единицу в триггер D1-1 канала арбитра, подключенного к абоненту. Если ОМ свободна, то единица с триггера D1-1 через элемент И записывается в триггер D1-2. При этом снимается сигнал обнуления счетчика D5 и рабочая частота, формируемая на выходе переноса, поступает на счетный вход счетчика D3. На линию ЗОМ, отображающую занятость магистрали, через элемент ИЛИ D8-1 выдается уровень занятости и запись единицы в триггер D1-2 через элемент ИД6 блокируется до освобождения ОМ. При снятии сигнала занятости с линии ЗОМ во всех каналах арбитра в счетчики D3 записывается число К—М, где К коэффициент пересчета счетчика, М — число абонентов.

Период импульсов рабочей синхрочастоты задается исходя из удвоенного времени распространения сигналов на ОМ. По каждому синхримипульсу с выхода счетчика D5 через мультиплексор D4 последовательно выдаются

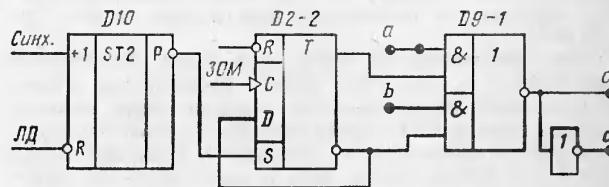


Рис. 2. Функциональная схема узла изменения приоритета

разряды кода приоритета X на общую линию ЛД. На схеме сравнения D6-2 выделяются каналы, с которых на ЛД выдается нулевой уровень. В этих каналах по сигналу переноса счетчика D5 в триггеры записываются нули, блокирующие дальнейшую выдачу разрядов кода приоритета соответствующих абонентов.

По сигналу переноса счетчика D3, вырабатываемому при завершении цикла выдачи всех разрядов кода приоритета, в единственном наиболее приоритетном канале, требующем захвата магистралей, единица из триггера D1-2 перепишется в триггер D2-1. При этом соответствующий абонент получает сигнал подтверждения захвата магистралей (ПЗМ).

Абонент, завершивший передачу пакета, выдает сигнал «Освободить ОМ» (ООМ). По этому сигналу в триггеры D1-1 и D1-2 записываются нули и с линии ЗОМ снимается уровень занятости магистралей. При этом во всех каналах, получивших требование захвата ОМ, в триггеры D1-2 записываются единицы.

Если необходимо изменять приоритеты абонентов после передачи каждого пакета, то к точкам a, b, c, d делается дополнительное включение (рис. 2). В этом случае по каждому сигналу на линии ЗОМ можно попеременно выдавать на ЛД прямой или инверсный код приоритета, что по существу реализует беспriorитетное обслуживание абонентов.

Телефон 555-24-40, Ленинград

ЛИТЕРАТУРА

1. Прангишвили И. В., Подлазов В. С., Стецюра Г. Г. Локальные микропроцессорные вычислительные сети. — М.: Наука, 1984. — 176 с.
2. А. с. 1383352 СССР. Многоканальное устройство для подключения абонентов к общей магистрали/В. А. Богатырев. — Оpubл. 1988. — Бюл. 11.
3. А. с. 1429132 СССР. Устройство для подключения абонентов к общей магистрали/В. А. Богатырев. — Оpubл. 1988. — Бюл. 37.

Сообщение поступило 25.01.89

УДК 681.3.06

В. Л. Динес, В. А. Мартюхин

СИСТЕМА УПРАВЛЕНИЯ ФАЙЛАМИ НА КНМЛ СМ5211

Кассетный накопитель СМ5211 имеет встроенный микроконтроллер, который управляется с помощью восьми специальных команд: записи блока данных и метки файла, чтения блока данных, пропуска блока или файла, возврата на блок или файл, перемотки.

Максимальная емкость кассеты 500 Кбайт, скорость обмена 1,25 Кбайт/с. Длина блока данных более 4 байт. Форматы записи метки файла и блока данных устанавливаются и проверяются на цикличность микроконтроллером накопителя. После выполнения каждой команды выдается код завершения операции.

КНМЛ подключается к КТС ЛИУС через адаптер, выполненный на двух микросхемах КР580ВВ55. Данные записываются на магнитную ленту в виде файла, ограниченного слева и справа метками и представляющего собой совокупность последовательных блоков данных (до 256 байт).

Файл вызывается по номеру или имени. Допускается организация любого числа файлов произвольной длины. Первый файл всегда содержит название (имя) стороны кассеты, пароль, дату и время создания; каждый последующий файл в первом блоке — специальную начальную часть: имя и тип файла, пароль, дату и время создания. Двухуровневая система паролей исключает несанкционированный доступ. Последним записывается специальный файл-маркер без имени и номера.

Комплекс программ управления данными на КНМЛ

включает монитор объемом 4 Кбайт и систему управления файлами (СУФ) объемом 4 Кбайт.

Номер накопителя КНМЛ присваивается логическим устройствам ввода-вывода (фотосчитывателю, перфоратору и печати) традиционным образом: левый накопитель имеет физическое имя (номер) 0, правый — 1.

При работе с СУФ различают два уровня команд — физический и логический. По командам физического уровня отлаживается и проверяется работоспособность КНМЛ. СУФ включает следующие команды физического уровня: записи блока и метки файла, чтения и пропуска блока, пропуска файла, возврата на блок, файл, перемотки. Вместе с командой вводятся дополнительные параметры, например номер накопителя, если он не зафиксирован командой FIX, и число пропускаемых блоков и файлов.

Команды логического уровня обеспечивают работу над файлом как единым целым. СУФ позволяет выполнять следующий набор операций: запись атрибутов стороны кассеты, обязательное начальное открытие кассеты, фиксацию рабочего номера накопителя, создание атрибутов файла, распечатку каталога файла, копирование кассеты, файла и перфоленты, сравнение копии и оригинала кассеты, файла, объединение нескольких файлов в один, установку на файл по имени или номеру, возврат на начало текущего файла, печать всего или части файла, уничтожение файлов, загрузку гексодесимального файла. Большинство команд сопровождается распечаткой меню, облегчающим взаимодействие пользователя с СУФ.

349870, Ворошиловградская обл., г. Рубежное, ул. Ленина, 31, ВЦ Рубежанского филиала Днепропетровского химико-технологического института; тел. 5-21-71, 7-30-12

Сообщение поступило 6.04.88

УДК 681.3.06

А. Ю. Черняк, И. П. Доровской, М. Л. Ланцузский, Н. Л. Прохоров

ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ ЦМД ЗУ

Операционная система (ОС) разработана с учетом специфических особенностей (архитектура, временные параметры) запоминающих устройств на цилиндрических магнитных доменах (ЦМД ЗУ) и предполагаемой области применения этой памяти (встроенные микропроцессорные системы повышенной надежности).

ОС состоит из набора модулей, реализованных на языке программирования Модуль-2: драйвера ЦМД ЗУ, файловой системы, загрузчика, обработчика исключительных ситуаций, модуля терминального ввода-вывода и монитора. Модули могут быть использованы совместно (в составе ОС) и (или) раздельно при создании пользователем прикладной программы.

Реализованная ОС достаточно компактна — версия для микроЭВМ «Электроника 60» занимает в оперативной памяти 8 Кбайт, в ЦМД ЗУ — 6,4 Кбайт (включая исполняющую систему Модуль-2). Она обеспечивает отображение, удаление, переименование и защиту, загрузку и выполнение пользовательских программ.

Для сравнения: ОС РАФОС — единственный альтернативный вариант для встроенных систем на базе микроЭВМ этого семейства — занимает в 12...15 раз больший объем внешней памяти, работает медленнее и менее надежно за счет использования оверлейс и свопинга, требует периодического «сжатия» носителя информации.

ОС ЦМД позволит быстро и качественно создавать малогабаритные микропроцессорные системы высокой надежности на основе одноплатных модулей ЦМД ЗУ (емкостью 64...256 Кбайт). Использование этой ОС в системах, оснащенных обычными накопителями на гибких магнитных дисках, обеспечит значительную экономию внешней памяти.

394693, Воронеж, Университетская пл., 1, ВГУ, ф-т ПММ, ЛВТ; тел. 56-СЗ-18

Сообщение поступило 23.01.89

М. В. Коллегов, В. В. Репков

ДИСПЛЕЙ С ВЫСОКИМ РАЗРЕШЕНИЕМ

Почти все известные растровые дисплеи как выпускаемые промышленностью, так и «самодельные» используют в качестве оконечного устройства отображения либо стандартные телевизоры, либо телевизионные мониторы со стандартной (телевизионной) разверткой. Принято считать, что телевизионный стандарт обеспечивает разрешение по строкам не менее чем 512 (625 полных строк, из которых примерно 50 строк занимает обратный ход). Однако при этом используется чересстрочная развертка и частота смены полей составляет 25 Гц.

Применение чересстрочной развертки оправдано при формировании «фотографического» изображения, т. е. там, где смежные строки раstra отображают аналогичную информацию. Для графического терминала с преобладанием контрастных деталей (тонкие горизонтальные и вертикальные линии) чересстрочная развертка практически не пригодна, так как приводит к недопустимому мерцанию изображения. Таким образом, телевизионный стандарт позволяет реализовать не более 260...280 строк немерцающего изображения. В этом случае чересстрочная развертка не используется, содержание обонх полукадров полностью совпадает и частота смены полей составляет 50 Гц. Обычно выбирается растр, состоящий из 256 строк с числом точек в строке в пределах 256...512.

В Институте ядерной физики СО АН СССР разработано несколько модификаций дисплеев, ориентированных на телевизионный стандарт, отличающихся числом цветов, интерфейсом связи с ЭВМ, функциональными возможностями. Эти дисплеи хорошо зарекомендовали себя в задачах автоматизации эксперимента, удовлетворительно — в задачах проектирования печатных плат. Однако при использовании их для представления чертежей, принципиальных схем и другой «чертежной» информации ограниченность информационной емкости экрана сказывается наиболее сильно. Поэтому разработан относительно дешевый дисплей, пригодный для графических применений.

Устройством отображения служит черно-белый монитор со следующими параметрами:

Размеры рабочего поля, чек	1024 × 768
Частота смены полей, Гц	50
Длительность одной строки, мкс	25 (прямой ход — 20 мкс, обратный ход — 5 мкс)
Полоса пропускания видеосигнала, МГц	50 (время высвечивания одной точки — 20 нс)

Для чертежных работ более важно, как правило, высокое разрешение по всему полю экрана, чем наличие цвета (много ли цветных чертежей?). В черно-белых кинескопах достигается более высокая четкость изображения и большая яркость, чем в цветных. Тем не менее, основные узлы монитора разрабатывались в расчете на их использование и в цветном варианте. Этому в значительной степени способствовали и основные технические параметры кинескопа.

Используется кинескоп 61ЛК8Б высокого разрешения. Колба кинескопа (стандартного цветного) с углом отклонения 90 град и 61 см по диагонали позволяет работать со стандартной отклоняющей системой ОС 90.38ПЦ12. Напряжение анода 25 кВ. Большое внимание уделено качеству фокусировки: имеется электрод «динамической фокусировки». Разрешение — не менее 200 линий по всему полю экрана.

Строчная развертка. Схема строчной развертки не содержит каких-либо принципиальных отличий от традиционной (рис. 1). Питание выходного каскада осуществляется через дроссель ДР1. Напряжение питания промодулировано «кадровой наработкой» для исправления подушкообразных искажений.

Основные особенности, связанные с более высоким значением строчной частоты:

напряжение питания увеличено с 50 до 130 В в связи с тем, что взята стандартная отклоняющая система, не обеспечивающая необходимой амплитуды отклоняющего тока;

пропорционально уменьшено время обратного хода луча, следовательно, возросла амплитуда выброса на транзисторе VT1 с 400 до 1100 В, а его длительность уменьшилась с 12 до 4,5 мкс.

Все это приводит к ужесточению требований к транзистору ключа; во-первых, он должен быть более высоковольтным, во-вторых, обладающим временем выключения не более 300 нс, в противном случае резко возрастает рассеиваемая мощность. Этим требованиям удовлетворяет транзистор КТ838.

При повышении частоты развертки улучшается линейность по горизонтали, поэтому не понадобился регулятор линейности. Объясняется это так: основным фактором, влияющим на линейность, является неидеальность ключа, которая приводит к возникновению «ступеньки» напряжения на ОС во время прямого хода строки. В первой половине строки напряжение определяется падением напряжения на диоде (примерно 1 В), во второй — напряжением насыщения транзистора (3...5 В). Легко показать, что нелинейность тока развертки пропорциональна отношению напряжения «ступеньки» к напряжению питания. Поскольку амплитуда «ступеньки» не изменяется, а напряжение питания возрастает пропорционально частоте, то линейность также улучшается. Правда, при этом не учитывалась нелинейность самой отклоняющей системы, но опыт показывает, что нелинейности такого типа практически отсутствуют.

Блок высокого напряжения. Принципиальная схема высоковольтного источника не содержит особенностей. Использовался стандартный трансформатор ТВС110П2 с перемотанной первичной обмоткой и умножитель типа УМ 8/25.

Кадровая развертка. Частота кадровой развертки не изменилась, поэтому использована стандартная схема, однако с некоторыми особенностями. В связи с тем, что амплитуда высоковольтного выброса на строчной ОС возросла почти в три раза, увеличилась наводка от строчной ОС в цепь кадровой ОС, что привело к искривлению горизонтальных линий в левой части экрана. Эффект может быть устранен, если кадровые катушки соединить параллельно. При этом несколько ухудшается режим выходного каскада, так как требуется вдвое больший ток при меньшем напряжении (рис. 2). Микросхема А1 — высоковольтный операционный усилитель К1408УД1. Выходной каскад представляет собой эмиттерный повторитель на комплементарных транзисторах. Во время обратного хода замыкается ключ К1 и напряжение питания выходного каскада становится равным 50 В, что обеспечивает быстрое переключение тока в кадровых ОС во время обратного хода.

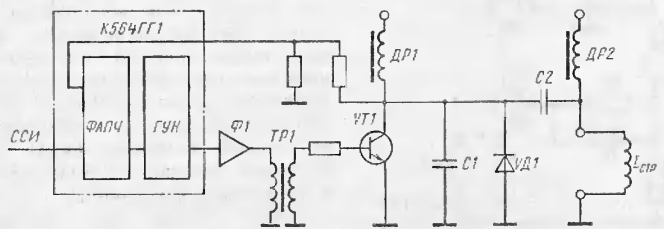


Рис. 1. Упрощенная схема строчной развертки

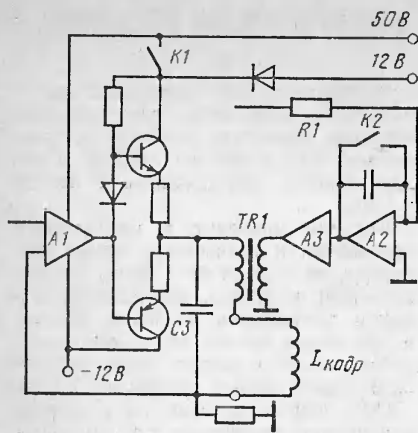


Рис. 2. Схема оконечного усилителя

Вторая особенность — исправление «подушки» по вертикали. Отметим сначала, что данный тип искажений может быть устранен соответствующей намоткой ОС и (или) применением корректирующих магнитов. Такая коррекция допустима для мониторов, имеющих кинескоп с планарным расположением пушек, т. е. в том случае, если вертикальный градиент магнитного поля не приводит к ухудшению сведения лучей или фокусировки. К сожалению, для данного типа кинескопов подобные ОС отсутствуют, поэтому коррекция осуществляется электронным путем.

Для коррекции необходимо, чтобы в кадровой ОС наряду с основным током присутствовал параболический, амплитуда и знак которого зависят от фазы «кадровой пилы». Следует отметить, что хотя добавка тока невелика (около 5%), но для напряжения строчной частоты кадровая ОС представляет собой чисто индуктивное сопротивление, поэтому для получения параболического тока на нее следует подать пилообразное напряжение, амплитуда которого примерно в 10 раз превосходит амплитуду напряжения кадровой развертки.

Для формирования пужного напряжения используется интегратор А2 (см. рис. 2). Ключ К2 замыкается строчными синхронимпульсами. К резистору R1 приложено напряжение «кадровой пилы». На выходе интегратора формируются пилообразные импульсы строчной частоты необходимой амплитуды и полярности. Усилитель А3 — буферный. Трансформатор TR1 повышает напряжение до необходимой амплитуды. Емкость C3 шунтирует напряжение строчной частоты в цепи кадровой развертки.

Видеоусилитель (рис. 3) должен обеспечивать формирование видеосигнала амплитудой до 30 В с минимальной длительностью импульсов 20 нс (время высвечивания одной точки) на емко-

сти модулятора 8 пФ. Для получения качественного изображения граничная частота видеоусилителя должна составлять примерно 30...40 МГц. Можно сформулировать задачу и несколько по-другому: видеоусилитель должен обеспечивать формирование фронтов видеосигнала длительностью примерно в одну треть от длительности высвечивания одной точки изображения (6 нс).

Для компенсации различия в яркости, связанной с конечным временем фронтов, полезно ввести небольшое переуправление, т. е. обеспечить подъем усиления на частоте примерно 30 МГц. Транзисторы VT1 и VT2 включены по каскодной схеме. Коэффициент усиления определяется отношением сопротивлений R1 и R2. Цепи R3, C3 и R4, C4 служат для коррекции АЧХ на высоких частотах. Цепочки VD1, VD2, VT3, VT4 осуществляет привязку к уровню черного и ограничивает напряжение на коллекторе VT1 на уровне 40 В. Транзистор VT1 типа KT919, VT2 — KT939B, номинал нагрузочного резистора R1 — 470 Ом.

Динамическая фокусировка. Одним из недостатков, объективно присущим кинескопам, является невозможность обеспечить равномерную фокусировку по всему полю экрана. Объясняется это тем, что радиус кривизны экрана много больше, чем расстояние от экрана до фокусирующей линзы. Обычно фокусировка выбирается так, чтобы добиться примерно одинакового качества по всему полю. Если этот эффект является основным, влияющим на качество изображения, то можно применять так называемую динамическую фокусировку, т. е. изменять фокусирующее напряжение в зависимости от отклонения луча от центра экрана. Как уже упоминалось, в кинескопе кроме основной фокусирующей линзы имеется дополнительная. Уровень напряжения на основном электроде устанавливается равным некоторому среднему значению. Тогда относительно небольшого изменения на электроде динамической фокусировки (~150 В) достаточно для получения оптимальной фокусировки по всему полю экрана.

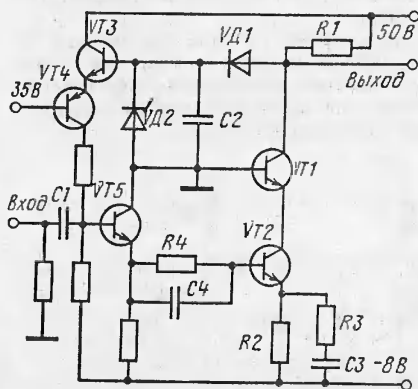


Рис. 3. Схема видеоусилителя

Из простых геометрических соображений следует, что фокусирующее напряжение должно быть пропорционально квадрату отклонения луча от центра экрана. Следовательно, напряжение динамической фокусировки должно представлять собой сумму параболических напряжений строчной и кадровой частот. Качество кинескопа позволяет на его основе сделать дисплей с разрешением порядка 2000×2000 точек.

Привод графического дисплея выполнен на одной двойной плате стандарта микроЭВМ «Электроника 60». Он обеспечивает вывод на экран 1024×768 точек растрового изображения с 16 градациями яркости или (при использовании цветного монитора) с 16 цветами. Привод содержит видеоЗУ со схемой формирования раstra, графический процессор, ЗУ выходной перекодировки цвета и интерфейс связи МПИ.

ВидеоЗУ (рис. 4) содержит четыре плоскости изображения размерами 1024×1024 бит, в которых используются микросхемы K565PV5 (по 16 микросхем на одну плоскость). Минимальное время считывания из этих микросхем 250...300 нс, темп вывода информации из видеоЗУ составляет 50 Мбит/с (время высвечивания одной точки 20 нс). Следовательно, недостаточное быстродействие микросхем ОЗУ требует их параллельного объединения для вывода растрового изображения.

Микросхемы одной плоскости могут быть объединены в линейный массив 16-разрядных слов. В такой организации для постановки одной точки изображения необходимо сначала прочитать слово из видеоЗУ, установить в нем соответствующий бит и записать обратно. Если необходимо записать точку сразу в нескольких плоскостях, то операцию надо проделать для каждой плоскости в отдельности.

Другой подход состоит в том, чтобы объединить информационные входы всех микросхем, а данные подавать на входы записи WE. Информация при этом «дописывается» в соответствующие микросхемы (записывается 0 или 1 в зависимости от состояния объединенного сигнала DI). Для графического дисплея такая организация оказывается более предпочтительной, так как при генерации изображения вдвое уменьшается число обращений к ЗУ и можно записать точку в нескольких плоскостях одновременно.

Каждая плоскость памяти состоит из двух блоков по восемь микросхем для четного и нечетного байтов. Циклы чтения этих байтов сдвинуты на половину цикла (рис. 5), а соответствующие выходы объединены попарно. Это допустимо, поскольку информация присутствует на выходах микросхем только во второй половине цикла чтения. В первой половине выходы находятся в неактивном состоянии.

Таким образом, при сканировании видеоЗУ на выходах присутствует мультиплексированный сигнал (снача-

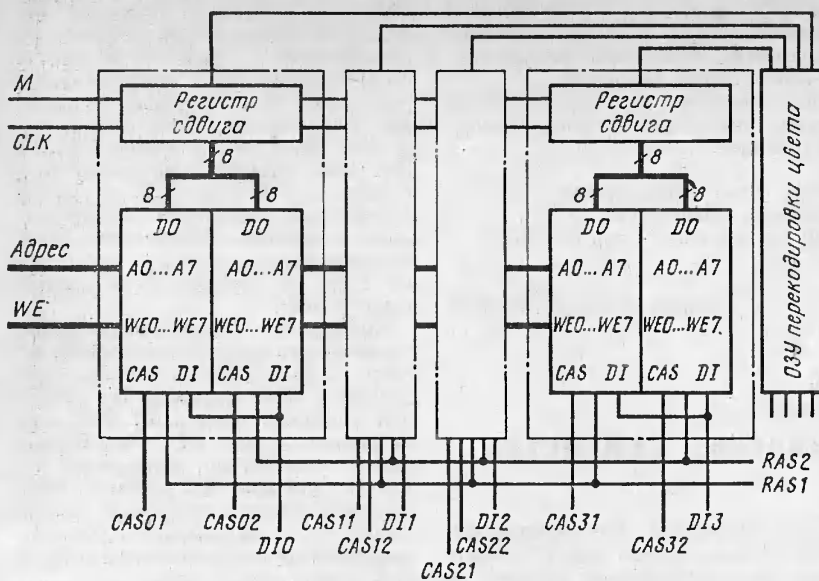


Рис. 4. Схема видеоЗУ

ла младший, потом старший байты). Это позволило вдвое сократить длину сдвиговых регистров, формирующих видеосигналы плоскостей.

Все информационные входы микросхем памяти каждой плоскости связаны между собой. Каждый разряд 8-разрядной шины WE объединяет соответствующие микросхемы во всех плоскостях и обоих блоках четного и нечетного байтов. Плоскость при записи в видеоЗУ выбирается подачей сигналов CAS на микросхемы выбранных плоскостей. Выбор четного или нечетного байта осуществляется синхронизацией циклов записи и обращения к четному или нечетному байту.

Такая организация позволяет за один цикл обращения реализовать режим записи как одной точки с произвольным цветом (записываются четыре бита информации в четырех плоскостях), так и восемь битов информации в строке.

Четыре плоскости памяти могут отображаться следующими способами:

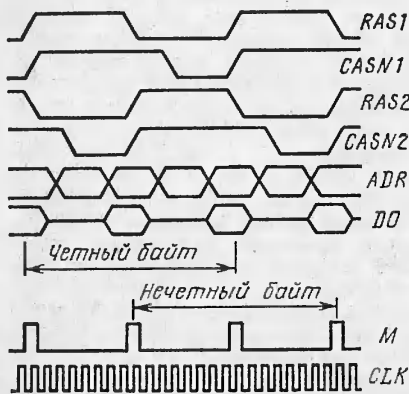


Рис. 5. Временные диаграммы циклов четного и нечетного байтов

поля расположены «рядом» и образуют одно общее поле размером $2048 \times 2048 \times 1$; или в режимах $1024 \times 2048 \times 2$, $2048 \times 1024 \times 2$, $1024 \times 1024 \times 4$. В последнем режиме одновременно выводятся 4 бит, что позволяет получить на цветном мониторе 16 цветов или 16 градаций яркости в черно-белом варианте.

Блок имеет режим «лупы» и «окна», т. е. на экран может выводиться часть изображения в увеличенном в два или четыре раза масштабе. Начальная координата окна определяется с дискретностью 16 точек. Предусмотрена возможность разрешать или запрещать обращение к видеоЗУ во время прямого хода луча. В первом случае скорость рисования выше, но появляется «рябь» на экране.

Вывод информации осуществляется через ЗУ цветности с организацией 4×16 бит (используется в режимах, где на одну точку изображения приходится более одного бита).

ЗУ перекодировки цвета позволяет установить произвольное соответствие между прочитанным из видеоЗУ кодом и уровнями видеосигнала на выходе. Благодаря этому имеется возможность осуществлять логические функции между плоскостями, задавать приоритеты цветов, маскировать фрагменты изображения, включать, выключать, инвертировать изображение некоторых плоскостей и т. д.

Графический процессор (ГЦ) представляет собой 12-разрядный микропрограммный автомат (рис. 6). АЛУ автомата выполнено на базе трех процессорных секций K1804BC1. Микропрограммы автомата хранятся в ПЗУ объемом 2048 команд. Микрокоманды автомата 24-разрядные. Время выполнения одной микрокоманды составляет 160 нс. По старшему биту они разделяются на два типа — операции с АЛУ и команды перехода. При опе-

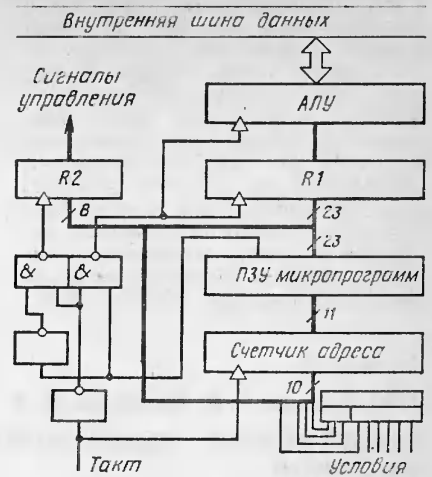


Рис. 6. Структурная схема графического процессора

рациях с АЛУ управляющие сигналы запоминаются в регистре R1 и происходит выборка следующей команды из ПЗУ. Ее адрес определяется счетчиком команд. При выполнении команды перехода содержимое регистра R1 не изменяется, подача такта на процессорных секциях блокируется, таким образом сохраняется результат выполнения предыдущей операции с АЛУ. Поле микрокоманды используется как адрес перехода и для управления селектором условий (младший разряд счетчика адреса загружается через селектор условий). В качестве условий могут использоваться флаги процессорных секций, кадровый и строчный синхронимпульсы (условные переходы). При безусловном переходе младший бит счетчика команд также берется из ПЗУ микрокоманд (подключается соответствующий вход селектора условий). Данная конфигурация позволяет загружать в счетчик команд адрес, младший бит которого определяется наличием соответствующего условия, и тем самым разветвлять микропрограммы.

Автомат поддерживает циклы обмена данными со всеми регистрами, имеющими выход на внутреннюю шину данных ГП: регистры координат, байта для записи точки в видеоЗУ, цвета, начальных координат окна вывода на экран, ОЗУ выходной перекодировки цвета и адресный регистр ПЗУ символов. ПЗУ символов содержит алфавит из 256 символов в матрице 8×8 точек. Предусмотрен режим чтения автоматом видеоЗУ, необходимый для сдвигов и закраски областей.

Плата привода занимает в адресном пространстве микроЭВМ «Электроника 60» около 20 адресов (зависит от набора микропрограмм). Обращение к каждому из них инициализирует соответствующую функцию процессора.

Данные передаются во время цикла непосредственно в одну из ячеек АЛУ ГП (или из нее).

ГП рисует точки, векторы, окружности, дуги, символы, прямоугольники. Действие всех команд может быть ограничено произвольной прямоугольной областью. Скорость рисования составляет 3 мкс на точку в случае, когда разрешена запись во время прямого хода. Область изображения, выводимая на экран, «стирается» аппаратным способом. Максимальное время выполнения операции стирания 40 мс.

Сдвиги фрагмента изображения. Сдвигается прямоугольная область изображения, задаваемая координатами концов любой диагонали.

Закраска областей. Закрашиваются области, ограниченные одной замкнутой границей.

630090, Новосибирск, проспект Академика Лаврентьева, 11, ИЯФ СО АН СССР; тел. 35-99-52, 35-95-63

Статья поступила 28.04.88

УДК 681.327.22

Н. Ю. Данилов, А. Д. Полукаров, М. В. Столяров

ДИСПЛЕЙНЫЕ МИКРОКОНТРОЛЛЕРЫ СЕМЕЙСТВА МИДИКОН

На базе микропроцессорного комплекта БИС К580 разработано семейство микроконтроллеров (МК) МИДИКОН, предназначенных для построения текстографических систем широкого применения. В зависимости от вида выводимой информации, конструктивного и схематехнического исполнения, технических возможностей они подразделяются на алфавитно-цифровые (текстовые), графические и комбинированные.

Семейство МК состоит из четырех базовых моделей: МГК-80, МЦД-90, МИДИКОН-80, МДК-80, выполненных на платах в конструктиве микроЭВМ «Электроника 60». Помимо базовых моделей имеются их модификации. Микроконтроллер МГК-80 (рис. 1) предназначен для оснащения АЦ-дисплеев с целью отображения графической информации на экране черно-белого или цветного монитора с телеви-

зионной разверткой. Он представляет собой функциональный блок с последовательными интерфейсами на входе и выходе для сопряжения с ЭВМ и любым стандартным АЦ-дисплеем. Конкретная реализация определяется выбранным конструктивным решением и способом согласования с синхросигналами разверток дисплея.

МГК-80 используется совместно с дисплеем «Электроника 15ИЭ-00-013», образуя комбинированный текстографический дисплей. МК устанавливается в двух свободных позициях блока логики дисплея и включается в разрыв линии связи между ЭВМ и АЦ-дисплеем (пунктирная линия на рис. 1 обозначает разрываемую связь). Цепь видеосигнала, проходящего от АЦ-контроллера к монитору, также разрывается, но при этом в МГК-80 графические и символические видеосигналы смешиваются и объединенный видеосигнал подается на монитор. В неграфическом режиме МГК-80 «прозрачен» для кодовых посылок и только определенные управляющие кодовые последовательности интерпретируются как графические команды, которые в АЦ-дисплее не передаются, а обрабатываются внутри МК. МГК-80 размещается на плате микроЭВМ и графической памяти. На первой расположен МП с памятью и узлами сопряжений, на второй — память и схема управления.

В МГК-80 используется растровый принцип отображения. Емкость графической памяти, выполненной на 16 микросхемах К565РУ5, составляет 128 Кбайт, что позволяет хранить матрицу изображения размерами $512 \times 512 \times 4$ точек. В схеме управления графической памятью предусмотрена возможность представления матриц изображений размерами $1024 \times 1024 \times 1$ и $1024 \times 512 \times 2$ точек и преобразования цветов (палитры). Максимальное число цветов или градаций яркости, одновременно воспроизводимых на экране, равно 16, полная гамма цветовых оттенков — 4096.

Ограниченные возможности приме-

няемых мониторов не позволяют выводить на экран полное изображение, размещаемое в графической памяти. МГК-80 обеспечивает просмотр графической памяти с помощью окна размером 400 точек по горизонтали и от 1 до 275 точек по вертикали. Положение окна относительно графической памяти, перемещение его и размер по вертикали изменяются соответствующими командами. Изображение может передаваться на экран с масштабом от 1 до 16 (режим лупы — репликация точек).

Предусмотрено подключение средств графического диалога типа трекбол или мышь. Характеристики канала связи дисплея с ЭВМ сохранены без изменения (скорость обмена 75...9600 Бод, интерфейс — стык С2 или Токовая петля). Управление графической памятью с помощью МП позволяет гибко изменять команды построения изображений, что в свою очередь, обеспечивает адаптируемость дисплея к конкретным прикладным задачам.

В частности, для МГК-80 разработаны две основные разновидности микропрограмм: одна реализует точечный принцип работы для решения графических задач; другая — мнемознаковый (псевдосимвольный) — для отображения мнемосхем технологических процессов.

Разновидностью МГК-80 является МК МГК-80/100, отличающийся типом микросхем памяти: К565РУ7 (256 Кбит на корпус), что позволяет хранить матрицу изображения размерами $1024 \times 1024 \times 4$ точек и использовать МК для решения задач САПР радиоэлектронной аппаратуры. С графическим дисплеем, построенным на базе МГК-80/100, успешно используется программа PED, представляющая собой графический редактор печатных плат [1]. Для работы с программой PED разработана специальная микропрограмма.

Микроконтроллер АЦД-80 рассчитан на совместную работу с МК МГК-80. Конструктивно АЦД-80 выполнен в виде одной платы размерами 135×240 мм. Встраивается в АЦ-дисплей «Электроника 15ИЭ-00-013» на место платы интерфейса ввода-вывода и полностью заменяет блок логики дисплея, который по объему эквивалентен шести платам АЦД-80. АЦД-80 функционально совместим с дисплеем VT-52 фирмы DEC (США), что позволяет использовать его в среде ЭВМ СМ1420, СМ4, «Электроника 60» и т. д. АЦД-80 обеспечивает вывод на экран до 80 символов в 24 строках, отображение дополнительного набора символов псевдографики, негативное изображение символов; реализует подчеркивание, мигание символов и другие возможности. Гарнитура символов имеет улучшенный вид за счет увеличения размера матрицы — 9×9 при знакоместе 12×12 .

Микроконтроллеры АЦД-80 и МГК-80 в сочетании с монитором и клавиатурой дисплея «Электроника 15ИЭ-00-013» образуют комбинированную тек-

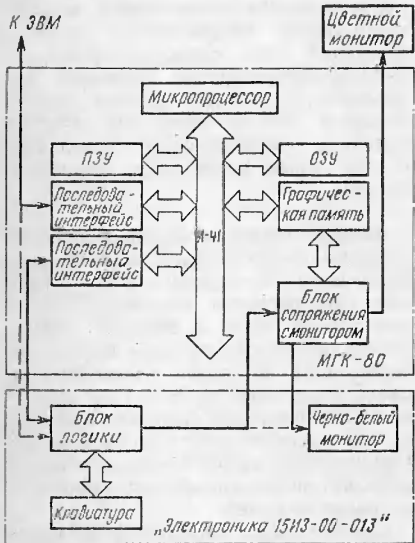


Рис. 1. Структурная схема графического микроконтроллера МГК-80

стографическую систему с экранным растром 480×288 точек.

Микроконтроллер МИДИКОН-80 предназначен для создания текстографических систем на базе микроЭВМ. Он представляет собой комбинацию МК МГК-80 и АЦД-80, но имеет единый общий МП и размещен на одной плате размерами 240×270 мм в конструктиве микроЭВМ «Электроника 60». Кроме того, в МК реализован параллельный интерфейс с выходом на МПИ, что позволяет встраивать его непосредственно в корзину микроЭВМ «Электроника 60» и использовать в качестве контроллера системного дисплея. По техническим характеристикам устройства МИДИКОН-80 в основном аналогичны МК МГК-80 и АЦД-80.

В МК МИДИКОН-80 используется расширенная таблица цветов. Предусмотрена возможность подключения монитора, клавиатуры, трекбола и адаптера связи с телевизионной камерой непосредственно через устанавливаемые на плате разъемы. На основе устройства МИДИКОН-80 созданы некоторые разновидности МК: МИДИКОН-80/1000 с параметрами, аналогичными контроллерам МГК-80/1000; МИДИКОН-80/К — МК МИДИКОН-80, выполненный в конструктиве КАМАК.

Микроконтроллер МДК-80 (текстографический) разработан для переноса серийно выпускаемых АЦД-дисплеев с целью улучшения их технических характеристик и расширения функциональных возможностей. Располагает буферной памятью регенерации экрана емкостью 640×256×1 бит, что дает возможность записывать и воспроизводить на экране не только знаки символов, но графические изображения по всему полю раstra.

Микроконтроллер МДК-80 (рис. 2), так же как и другие из семей-

ства МИДИКОН, построен на базе МПК БИС серии К580, ОЗУ — на микросхемах К565РУ6 общим объемом 32 Кбайт. Из них 20 Кбайт используется для графического буфера или памяти экрана дисплея. Последняя по способу обращения из микропрограммы организована в виде матрицы байтов 80×256, а при отображении — в виде матрицы точек 640×256×1. Оставшиеся 12 Кбайт составляют рабочую область памяти МП. ПЗУ объемом до 32 Кбайт предназначено для хранения микропрограммного обеспечения дисплея.

Дисплей на основе МДК-80 функционально совместим с дисплеем VT-52 фирмы DEC (США) и дисплеем 4010 фирмы TEKTRONIX (США).

Конструктивно МК МДК-80 выполнен на одной плате микроЭВМ «Электроника 60» размерами 240×270 мм. На ней можно реализовать два варианта интерфейсов: один для встраивания МК в дисплей «Электроника 15ИЭ-00-013» (при этом он полностью заменяет блок логики дисплея); другой для подключения к шине МПИ и связи через специальные разъемы с видеомонитором (например, МС6105) и последовательной клавиатурой (например, МС7004). При таком использовании МК МДК-80 по отношению к микроЭВМ «Электроника 60» представляет собой стандартный интерфейс в виде четырех регистров на странице ввода-вывода ЭВМ. Разработан вариант МК МДК-80, предназначенный для модернизации широко распространенных дисплеев VDT-52100 производства ВНР.

Реальные графические дисплеи и текстографические системы, созданные на основе МК семейства МИДИКОН, в настоящее время работают в составе ЭВМ СМ4, СМ1420 и «Электроника 60» с использованием графических пакетов Р-300, PLOT-10, ГРАФОР, АТОМ-84 [2], графических диалоговых систем

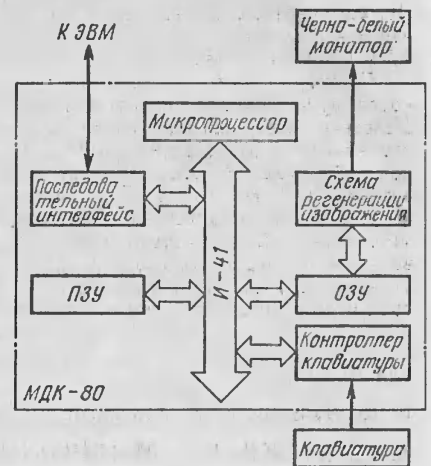


Рис. 2. Структурная схема дисплейного микроконтроллера МДК-80

PLOT и SURF, DISCORE/GRAFOR [3].

Телефон 196-94-75, Данилов Николай Юрьевич, Москва

ЛИТЕРАТУРА

1. Галныкин Э. А. Система проектирования печатных плат на ЭВМ семейства «Электроника» // Автоматрия. — 1984. — № 14.
2. Кочин В. Н., Самарин А. В. Графический пакет АТОМ-84. Функциональная характеристика. — Препринт ИФВЭ 85-137. — Серпухов, 1975.
3. Пиковский А. С. Состояние и перспективы развития графической системы // Тез. докл. II Всесоюз. конф. «Методы и средства обработки сложной графической информации». — Горький, 1985.

Статья поступила 14.01.88

УДК 681.3

А. О. Ладис, Вс. С. Штаркман

СИСТЕМЫ КОМАНД РАСТРОВОГО ГРАФИЧЕСКОГО ДИСПЛЕЯ «ЭЛЕКТРОНИКА МС7401»

Все выполняемые дисплеем (рис. 1) функции по отображению информации на экране реализованы программно на встроенной ЭВМ. Видеопамять прямо адресуется процессором этой ЭВМ. Внешними устройствами встроенной ЭВМ является клавиатура и линия связи с основной ЭВМ. Таким образом, система команд дисплея, «видимая» для основной ЭВМ, определяется управляющей программой, «защитой» в ПЗУ встроенной машины дисплея.

Основные точки принятия проектных решений

Совмещение алфавитно-цифровых и графических возможностей. Дисплей ориентирован на применение в качестве единственного терминала пользователя (возможно, даже единствен-



Рис. 1. Структурная схема дисплея

ного терминала ЭВМ), т. е. он должен быть одновременно алфавитно-цифровым (АЦ) и графическим.

Дисплей предназначен для ЭВМ типа ДВК—СМ4. Прототипом АЦ системы команд (как старой*, так и новой) была выбрана система команд терминала VT-52 фирмы DEC с кодировкой 7 бит (используется 7 битов в каждом байте). «Специальные функции управления», к которым в нашем случае относится и вся графика, кодируются в виде ESC-последовательностей. Символ ESC — префикс специальной функции, следующий символ — код функции, затем, если требуется, идут параметры.

Совмещенный или графический режим? Возникает вопрос, как обеспечить сосуществование АЦ и графического потоков информации? Должен ли терминал переключаться некоторой ESC-последовательностью в чисто графический режим, когда АЦ функции становятся недоступными, или должен быть обеспечен совмещенный режим, когда каждая графическая команда кодируется отдельной ESC-последовательностью?

Многие зарубежные разработчики [1] при создании совмещенных дисплеев аналогичного класса предусматривают графический режим как единственную возможность воспользоваться графикой. Это дает компактность кодирования графической информации и простоту ее декодирования в дисплее.

Опыт разработки базовых программных средств графики выявил серьезные недостатки такого подхода. Прикладную программу для совмещенного дисплея естественно писать в расчете на два «независимых» логических устройства: АЦ и графический дисплей. При этом программист должен заботиться, чтобы физически этим двум устройствам соответствовали разные области на экране. Однако было бы неестественно требовать от программиста еще и помнить, в какой момент на какое устройство надо переключиться. Хотелось бы, чтобы наряду с возможностью рисовать картинку прикладная программа могла в любой момент что-то напечатать на терминале в обычном АЦ режиме. Ведь момент АЦ печати не всегда легко определить: какая-либо стандартная подпро-

грамма, не имеющая отношения к графике вообще, может «неожиданно» выдать на терминал, к примеру, сообщение об ошибке, которое в графическом режиме, конечно же, воспримется как непредсказуемая последовательность графических команд.

Проблема эта возникает из-за того, что типичные ОС тех ЭВМ, с которыми предполагалось в основном использовать дисплей, не рассчитаны на «совмещенное» устройство, «ничего не знают» про графику. В этих условиях более приемлемым является совмещенный режим, в котором каждая графическая команда выделяется дисплеем из единого потока АЦ символов и графических команд.

Выбор способа кодирования для графики. При использовании совмещенного режима остро стоит проблема компактности кодирования графических команд: одно из узких мест аппаратуры — низкая скорость линии связи дисплея с основной ЭВМ. Смешанный поток вывода предполагает, что каждая графическая команда помечена префиксом, который является «накладным расходом» и должен быть как можно короче.

Далее, разумно требовать, чтобы дисплей управлялся терминальным драйвером ОС, рассчитанным на обычный АЦ терминал. В то же время для представления графической информации выгодно из соображений компактности использовать двоичное кодирование. Однако известно, что терминальные драйверы многих ОС, во-первых, используют 7-бит код, а во-вторых, не обеспечивают «прозрачной» передачи, т. е. трактуют некоторые 7-бит коды как управляющие, не передают их без изменений устройству.

Здесь возможны два выхода: передавать графические команды в АЦ виде или перейти к 6-бит коду. При 6-бит кодировании появляется возможность старший из семи разрядов 7-бит кода устанавливать в «1», «маскируя» произвольный 6-бит код под изображаемый АЦ символ. Это решает проблему прозрачности, поскольку изображаемые символы пропускаются любым драйвером без изменений.

Функциональные возможности графической системы команд следует сопоставлять прежде всего со скоростными свойствами аппаратуры. Рассмотрим с этой точки зрения простейшую для растрового устройства возможность — пересылку копии видеопамяти из основной ЭВМ в дисплей. Размеры экрана — 560×400 точек, формат видеопамяти — 7 точек на 1 байт, что дает общий объем видеопамяти около 32 Кбайт. Пересылка всего экрана по линии с быстродействием 9600 Бод займет около 30 с, по линии 2400 Бод — почти 1,5 мин. Все это заставляет искать пути интеллектуализации графической системы команд.

* Дисплей серийно выпускается промышленностью с 1983 г. Сотрудниками предприятия-изготовителя А. И. Хариним и Ю. Ф. Костровым была разработана и реализована система команд, с которой дисплей поставляется в настоящее время. Эту систему команд будем далее именовать «старой», в отличие от описываемой «новой». При разработке «старой» системы команд за основу был принят эскизный проект, выполненный В. А. Димовым и А. О. Лаисом.

На этапе разработки старой системы команд были в основном найдены ключевые точки принятия проектных решений. Однако недостаток опыта у разработчиков привел к тому, что некоторые из этих ключевых решений были приняты не лучшим образом. Эксплуатация старой системы команд выявила ряд серьезных недостатков как по замыслу, так и по исполнению. Поэтому решено основательно ее пересмотреть.

Новая система команд создана в ИПМ АН СССР в 1986 г. при содействии коллектива разработчиков старой системы команд. Цель этой статьи — проанализировать основные точки принятия проектных решений в старой и новой системах команд.

Векторный вход. Сократить объем передаваемой в дисплей информации можно, реализовав «векторный вход»: основная ЭВМ передает в дисплей команды «нарисовать отрезок» (дугу, окружность...), а встроенная ЭВМ дисплея разлагает передаваемые ей графические примитивы в растр. К векторным можно добавить растровые примитивы (т. е. закраска областей).

Рассмотрим скоростные соотношения на примере команды «нарисовать отрезок». Параметрами команды являются две координаты (отрезок проводится из текущей позиции в указанную). Для кодирования координаты требуются 2...4 байт (при использовании самой компактной — двоичной — кодировки), еще 2 байт — код операции и префикс. Таким образом, длина команды оценивается в 4...6 байт.

Какова скорость построения отрезка? Для построения отрезков в новой системе команд был выбран алгоритм Брезенхема [2]. Тщательная оптимизация программы позволила получить скорость критического цикла алгоритма около 16 000 точек/с.

К сожалению, расходы на построение отрезка, не зависящие от его длины (распаковка графической команды и подготовка критического цикла), составляют эквивалент времени построения 130 точек. При рисовании отрезков длиной 100 точек это дает эффективное быстродействие немногим более 5000 точек/с. Очевидно, что при АЦ кодировании на задание отрезка при самом компактном формате команды уйдет 8...10 байт, время декодирования возрастет.

Из полученного соотношения скоростей следует, что при скорости передачи 2400 Бод основным «узким местом» является линия. При скорости 9600 Бод сдерживает в основном скорость рисования, если не говорить о случае большого числа коротких (20...30 точек) отрезков. Каждое из этих двух «узких мест» может быть расширено.

Выборочное стирание могло бы резко повысить динамику изображения, несмотря на довольно низкую скорость рисования. Простейший способ предоставить основной ЭВМ возможность выборочного стирания — хранить в дисплее изображение в виде совокупности графических примитивов (в том виде, в каком его присылает главная ЭВМ). Выборочное стирание в этом случае осуществляется в хранимой картинке, а задача встроенной ЭВМ дисплея — поддерживать соответствие между хранимой картинкой и содержимым видеопамати, всякий раз заново разлагая хранимую картинку в растр. Такая схема очень похожа на работу векторного дисплея с аппаратной регенерацией изображения. Однако у векторного дисплея любое изменение хранимой картинки вызывает мгновенное изменение изображения на экране. В на-

шем случае мгновенного изменения картинки получить не удастся. Значит, надо уметь стереть часть картинки, физически не трогая всего остального.

Стирающее и инвертирующее перо — способ организовать выборочное стирание на растровом дисплее. В зарубежных дисплеях аналогичного класса и в старой системе команд такие средства есть. Проще всего обеспечить выборочное стирание фрагментов, геометрически не пересекающихся с остающейся частью изображения: достаточно реализовать, наряду с рисованием белой краской, рисование черной (стирающее перо). Однако при стирании таким способом фрагмента, геометрически пересекающегося с остающейся частью изображения, в местах охлаждения «стирающего пера» на изображении остаются просветы. При многократном стирании изображение становится «изъеденным». Эта проблема имеет довольно элегантное решение. Наряду с рисующим и стирающим пером вводится перо инвертирующее, которое меняет состояние точек экрана на своем пути на противоположное. В этом режиме в точках пересечения четного числа отрезков образуются «выколотые» точки. Но они не размножаются, поскольку повторная прорисовка фрагмента инвертирующим пером ведет к его стиранию с «зарастанием» просветов. Все три режима (рисующее, стирающее и инвертирующее перо) присутствовали в старой и были сохранены в новой системе команд.

Хранение векторной картинки. Мы установили, что для имеющих скорость аппаратуры запоминание в дисплее изображения в виде совокупности графических примитивов в сочетании с возможностью нарисовать это изображение на экране по запросу основной ЭВМ не решает проблемы динамики изображения. Сам по себе этот прием является полезным, поскольку он экономит ресурсы основной ЭВМ и позволяет расширить второе «узкое место», связанное со случаем, когда рисование по тем или иным причинам обгоняет передачу.

Старая система команд

Совмещение алфавитно-цифровых и графических возможностей. По АЦ функциям старая система команд совместима с VT-52 и содержит ряд расширений. Наиболее полезна возможность ограничивать зону АЦ диалога произвольным прямоугольником знакомест на экране. Это позволяет избежать порчи графического изображения при протяжке АЦ экрана. К сожалению, в реализации этого режима допущена ошибка: в ограниченной по ширине зоне диалога протяжка работает неправильно, часть строк пропадает.

Есть и совмещенный, и графический режимы. Префикс графической команды имеет вид

ESC F (два символа). Одна из графических команд включает чисто графический режим, в котором префикс можно опускать. В каждом режиме допустимы два способа кодирования: десятичный и двоичный. При десятичном кодировании числовые параметры от кода команды и друг от друга отделяются пробелами. При двоичном кодировании параметры передаются в фиксированном формате в виде последовательности байтов, каждый из которых представляет собой произвольное 7-разрядное двоичное число. В этом случае разделители между параметрами отсутствуют, но код операции требуется отделять от параметров хотя бы одним пробелом. Признаком конца команды является символ «в. к.», «п. с.» или любая их комбинация, но в случаях, не оговоренных особо, признак конца команды — начало следующей.

Перечисленные здесь соглашения о кодировании графики, как показала практика, — это одно из самых слабых мест старой системы команд. Прежде всего, 7-бит кодировка двоичной информации не учитывает проблемы прозрачности передачи, что сразу же делает двоичное кодирование практически малоприменимым на некоторых ОС. Далее, двоичное кодирование недостаточно компактно: нет коротких (в 1 байт) параметров, имеется разделитель кода операции и параметров, несмотря на фиксированный формат команды. Как будет показано ниже, префикс графической команды можно совместить с кодом самой команды, сократив длину команды в совмещенном режиме еще на 1 байт. Соглашение о признаке конца команды просто неудобно: если программа в двоичной кодировке, когда формат команды фиксирован и признак конца не нужен, выдаст след за графической командой АЦ «в. к.», «п. с.», этот стандартный перевод строки не обработает как АЦ, а будет «приклеен к хвосту» графической команды.

Функциональные возможности графической системы команд. Координаты в командах рисования примитивов (отрезки, окружности, дуги и строки текста) задаются в целочисленной системе координат экрана (в точках растра). Можно пользоваться абсолютными и относительными координатами. Диапазон координат, допустимых в командах рисования, превышает размеры экрана. При рисовании за пределами экрана видимое изображение отсекается по краю экрана или по произвольному прямоугольному окну со сторонами, параллельными сторонам экрана. Выключить отсечение нельзя, что замедляет рисование. В то же время, если дисплей подключен к ЭВМ с существенно большим быстродействием, чем встроенная ЭВМ дисплея, отсечение было бы явно выгоднее выполнять в основной ЭВМ. Кроме того, во

многих прикладных задачах, решаемых на мало мощной основной ЭВМ, лучше следить на прикладном уровне за тем, чтобы ничего отсекалось не требовалось, чем снижать скорость рисования за счет аппаратного отсека.

Реализовано семь фиксированных типов линии, два размера литер графического текста и восемь направлений строки графического текста. Тип линии можно как установить отдельной командой, так и задать дополнительным байтом в командах «отрезок», «дуга», «окружность». Эта возможность себя не оправдывает, поскольку увеличиваются накладные расходы на дешифрацию соответствующих команд. Имеются четыре пера: рисующее, стирающее, инвертирующее, принудительно рисующее (для несплошных линий светлые точки рисовать, темные — стирать).

Схема работы с хранимой векторной картинкой — одно из самых неудачных мест старой системы команд. Это относится и к основным идеям, заложенным на этапе эскизного проектирования, и к ошибкам реализации. Вместо того, чтобы спроектировать схему работы с хранимой картинкой для данного класса аппаратуры, разработчики эскизного проекта пошли по пути косметической доработки схемы, хорошо зарекомендовавшей себя для векторных дисплеев с регенерацией, а именно схемы с сегментированным дисплейным файлом. Например, вместо команды «обновить экран» была сделана команда «выполнить сегмент». Предполагалось, что выполнение сегмента инвертирующим пером будет играть ту же роль, что гашение/зажигание сегмента на векторном дисплее с регенерацией. При этом фактически выпала из поля зрения весьма существенная деталь.

Изготовлением и прорисовкой дисплейного файла теперь, в отличие от случая векторного дисплея с регенерацией, занимается один процесс. Он характеризуется переменными состояниями: текущими позицией на экране и типом линии и т. п. Как же соотносятся между собой состояния процессов при изготовлении дисплейного файла и при его выполнении? Этот вопрос в проекте практически не был проработан. Поведение дисплея при работе с дисплейным файлом оказалось запутанным, лишеным логической прозрачности.

Дело еще более осложнилось возможностью применять сегментированный дисплейный файл с подпрограммами. Для случая применения векторного дисплея с регенерацией различие между сегментами и подпрограммами прозрачно: подпрограммы — просто сокращенная форма записи, в то время как сегменты — способ организовать динамику картинки и манипуляции с частями изображения. В старой же системе команд различие между сегментом и подпрограммой логически не очевидно, использование двух совпадающих поня-

тий вместо одного дезориентирует пользователя.

К сильным сторонам старой системы команд следует отнести возможность независимых манипуляций с экраном («выполнить сегмент», «стереть экран») и с дисплейным файлом («удалить сегмент», «сброс дисплейного файла»). Существенный недостаток — отсутствие закраски областей. Возможности старой системы команд не исчерпываются разобранными здесь, но, на наш взгляд, все остальные гораздо менее существенны.

Недостатки реализации. Реализация старой системы команд, к сожалению, также оказалась далеко не безупречной. Терминал временами «зависает». На некоторых экземплярах проявляется эпизодическое «двоение» принимаемых от ЭВМ символов. Настройка параметров дисплея (работа со служебной строкой) сложна и не наглядна. В реализации работы с дисплейным файлом есть серьезнейшая ошибка: удаление сегмента, даже последнего, не освобождает занятое им место.

Новая система команд

Совмещение алфавитно-цифровых и графических возможностей. Каждая графическая команда кодируется отдельной ESC-последовательностью, и, тем самым, префикс графической команды сокращен до одного символа ESC. Замечено, что символы старшей четверти 7-бит кодовой таблицы (начиная с 140 В) не используются в прототипе в качестве кодов функций ESC-последовательностей. Эти коды, соответствующие малым латинским буквам и некоторым спецсимволам, заняты под коды функций графических команд. Все эти символы — изображаемые и пропускаются стандартным терминальным драйвером.

Предусмотрен единственный способ кодирования параметров графических команд — двоичный 6-бит. Все команды, кроме одной («строка графического текста»), имеют фиксированный формат задания параметров без всяких разделителей и признаков конца. Для наиболее часто используемой команды («отрезок/сдвиг») предусмотрен короткий (1-байт) формат координат, что позволяет при рисовании короткими штрихами сократить длину команды «отрезок» до 4 байт.

Функциональные возможности графической системы команд. Реализован векторный вход и некоторые специфически растровые возможности (см. ниже). Набор графических примитивов — отрезки и строки текста; не реализованы окружности и дуги, нет также выбора шрифта и направления строки для графического текста. Нельзя рисовать за пределами экрана, нет и отсечения по окну, как нет и самого понятия окна. Отказ от отсечения позволил поднять ско-

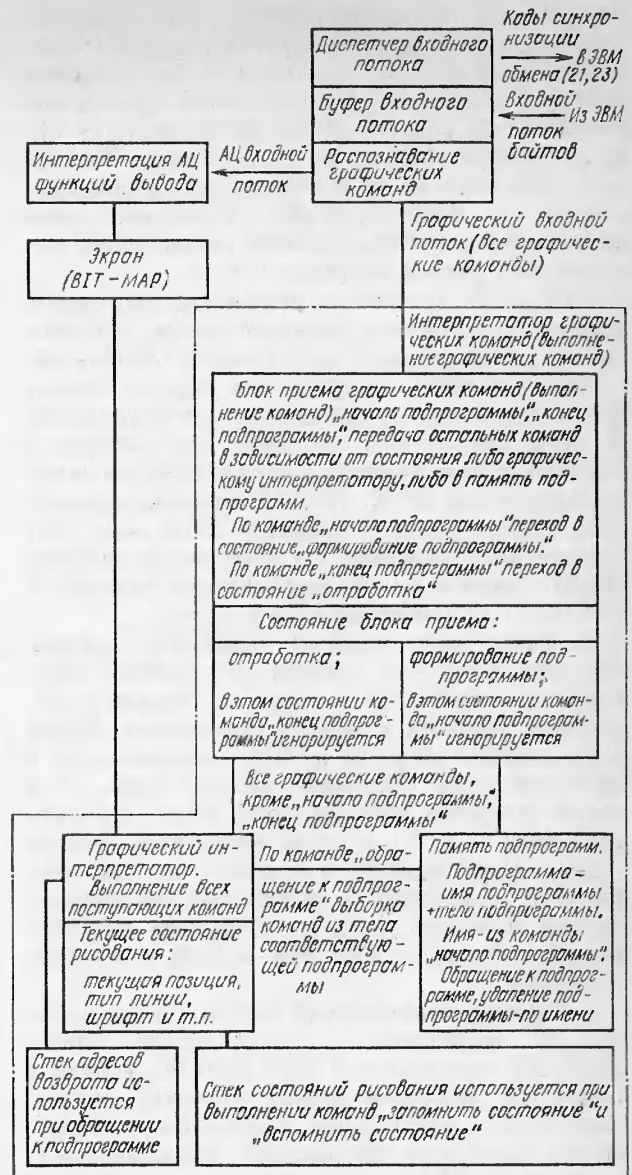


Рис. 2. Структура MC7401 (вывод)

рость рисования в 2—2,5 раза по сравнению со старой системой команд.

Заранее заданных типов линии, кроме сплошной, нет. Тип линии — задаваемый пользователем 8-бит код (параметр команды «установить тип линии»). Рисование несплошной линии заключается в том, что заданный в качестве типа линии 8-бит код размножается, как шаблон, вдоль рисуемого контура, причем рисуются (стираются, инвертируются) только те точки, которые соответствуют разрядам «1» в шаблоне. Более того, специальный бит в коде команды «отрезок» управляет тем, надо ли перед началом рисования данного отрезка установить шаблон на начало или следует продолжать с текущего состояния шаблона. Это позволяет плавно распространять произвольный тип линии

по ломаной, состоящей из сколь угодно мелких звеньев; кроме того, выборочно стирать с одинаковой легкостью сплошные и несплошные контуры, поскольку при повторной прорисовке несплошного контура легко обеспечивается попадание пера строго в те же точки. Можно выбирать одно из трех перьев: рисующее, стирающее и инвертирующее. Принудительная прорисовка несплошных линий реализована как отдельный режим рисующего пера.

Работа с хранимой векторной картинкой упрощена. Сегменты ликвидированы, остались только графические подпрограммы. Любая последовательность графических команд может быть объявлена графической подпрограммой. В этом случае она по мере поступления в дисплей не выполняется, а запоминается в памяти встроенной ЭВМ. Послав команду «выполнить подпрограмму», основная ЭВМ запускает выполнение запомненной в ЭВМ подпрограммы. Подпрограммы можно динамически удалять, и они могут обращаться друг к другу.

В явном виде понятия «параметр подпрограммы» в системе команд нет. Неявно передаваемым параметром является, прежде всего, текущая позиция в момент обращения. Кроме того, имеется мощный аппарат запоминания и восстановления состояния дисплея (рис. 2) в целом (текущей позиции, типа линии, выбранного пера и т. п.) в стеке состояний. Работа с ним осуществляется с помощью команд «запомнить состояние» и «вспомнить состояние». Будучи полезной сама по себе, эта возможность особенно важна при работе с подпрограммами.

Растровые возможности идейно заимствованы из графических возможностей языка БЕЙСИК персональной ЭВМ IBM PC. Имеется очевидная аналогия между проектированием графических расширений БЕЙСИКа и графических возможностей дисплея, подключенного к основной ЭВМ через медленную линию: в обоих случаях присутствует сходное «узкое горло» (в первом случае — потери на интерпретацию, во втором — низкая скорость передачи).

Предоставлены только две растровые возможности: режим закраски «прямоугольник» и работа с экраным буфером.

В режиме закраски «прямоугольник» при выполнении команды «отрезок» происходит не рисование отрезка, а закрашка (текущим пером) прямоугольника со сторонами, параллельными осям координат, для которого заданный отрезок является диагональю. Очевидно, имело смысл ввести еще режимы закраски, такие, как в графическом протоколе REGIS фирмы DEC [1], т. е. уметь залить произвольную трапецию (от указанного отрезка до заданной вертикали/горизонтали), но такая возмож-

ность не реализована. Это серьезный недостаток.

Работа с экраным буфером, т. е. возможность одной командой скопировать произвольный прямоугольник с экрана в ОЗУ встроенной ЭВМ и потом, при необходимости, восстановить его на экране. Емкость экранного буфера — примерно 0,5 экрана. Скорости выполнения растровых команд: заливка прямоугольников — около 1,5 экрана/с, работа с экраным буфером — 0,7 экрана/с.

Загрузка фрагментов управляющей программы из основной ЭВМ. В новой системе команд можно загружать во встроенную ЭВМ дисплея фрагменты управляющей программы из основной ЭВМ. Речь не идет о том, что пользователь дисплея мог бы писать свои произвольные программы для выполнения на встроенной в дисплей ЭВМ: такому пользователю пришлось бы слишком много знать об особенностях реализации уже имеющейся в дисплее программы. Однако наличие в принципе этой возможности весьма выгодно для разработчика, позволяет распространять мелкие доделки старым пользователям без перепрограммирования ПЗУ, а также организовать набор сменных компонентов, написанных разработчиком и загружаемых по выбору пользователя. Именно так могли бы быть реализованы, например, окружности, локально перемещаемое по экрану перекрестье, знакогенератор, отличный от имеющегося, и многое другое. Конечно, при желании можно использовать загружаемую из основной ЭВМ программу как «вторичный загрузчик» и полностью перепрограммировать встроенную ЭВМ по своему усмотрению.

Авторы считают своим долгом поблагодарить В. А. Димова за плодотворную дискуссию по эскизу старой системы команд, В. Н. Бершакова, Ю. Ф. Кострова, В. М. Радзиковского, А. И. Харина — за всемерное содействие в работе, неоднократные ценные консультации, а также за предоставление необходимой инструментальной аппаратуры, К. К. Мануйлова — за интерес к работе, ценные советы и помощь в прошивке опытных образцов ПЗУ с новой системой команд.

Телефон 333-80-55, Москва

ЛИТЕРАТУРА

1. VT-240 Series Programmer's Reference Manual. Digital Equipment Corp. EK-VT-240—RM-002, 1984.
2. Фоли Дж., ван Дэм А. Основы интерактивной машинной графики. — М.: Мир, 1985.
3. Уокер Б. С., Гурд Дж. Р., Дроник Е. А. Интерактивная машинная графика. — М.: Машиностроение, 1980.
4. Безобразов В. С., Димов В. А. и др. Контроллер графического дисплея // Микропроцессорные средства и системы. — 1986. — № 3.

Статья поступила 24.11.87

А. А. Бучнев, В. Г. Сизых, В. Ф. Минин

ЦВЕТНАЯ ПОЛУТОНОВАЯ СТАНЦИЯ ПРОФЕССИОНАЛЬНОГО НАЗНАЧЕНИЯ ГАММА-7.1

Дисплейная станция ГАММА-7.1 класса H₁F₁P₁ предназначена для использования в системах автоматизации проектирования любого профиля, при решении задач обработки изображений, математического моделирования, управления технологическими процессами и др., а также в качестве персональной ЭВМ.

Основные технические параметры

Экранное разрешение, дискрет	1024×768 для прогрессивной развертки монитора с частотой кадров 50 Гц; 1024×1024 для чересстрочной развертки монитора при 30 Гц
Общий размер изображения, пикселей	1024×2024
Объем видеопамати, Мбайт	1
Число градаций яркости или цветов	256
Число возможных цветов и оттенков	2 ²⁴
Минимальное время отображения пикселя, нс	20
Максимальный темп видеовывода, Мбайт/с	50
Скорость генерации векторов, нс/пиксель	640
Время генерации символа (литерная площадка 7×9 дискрет), мкс	100
Скорость стирания одной или всех плоскостей видеопамати или скорость раскраски больших площадей, нс/пиксель	80
Средства диалога	клавиатура и трекбол
Число аппаратных масштабов	4
Тип интерфейса с главной ЭВМ	любой в стандарте аппаратных средств шины МПИ

Дисплейная станция (ДС) выполнена по современной двухпроцессорной архитектуре, позволяющей разделить и распараллелить процессы обработки высокоуровневого представления графических данных и генерации их растрового представления, повысив тем самым интерактивность дисплейной системы в целом (см. рисунок).

В функции дисплейного процессора (ДП) входят прием, хранение, генерация, видовые преобразования и передача структурированного дисплейного файла. Полученная в результате видовых преобразований линейная последовательность графических примитивов наряду с другими директивами протокола «Дисплейный процессор — графический процессор» образует программу, предназначенную для исполнения графическим процессором (ГП). Другими функциями ДП являются: обеспечение символьно-графического

диалога, обслуживание периферийных устройств, поддержка взаимодействия с главной ЭВМ, исполнение прикладной программы. В качестве ДП может использоваться любая микроЭВМ универсального назначения, имеющая магистраль МПИ («Электроника МС1201.02», «Электроника МС1211», «Электроника МС1212» и др.).

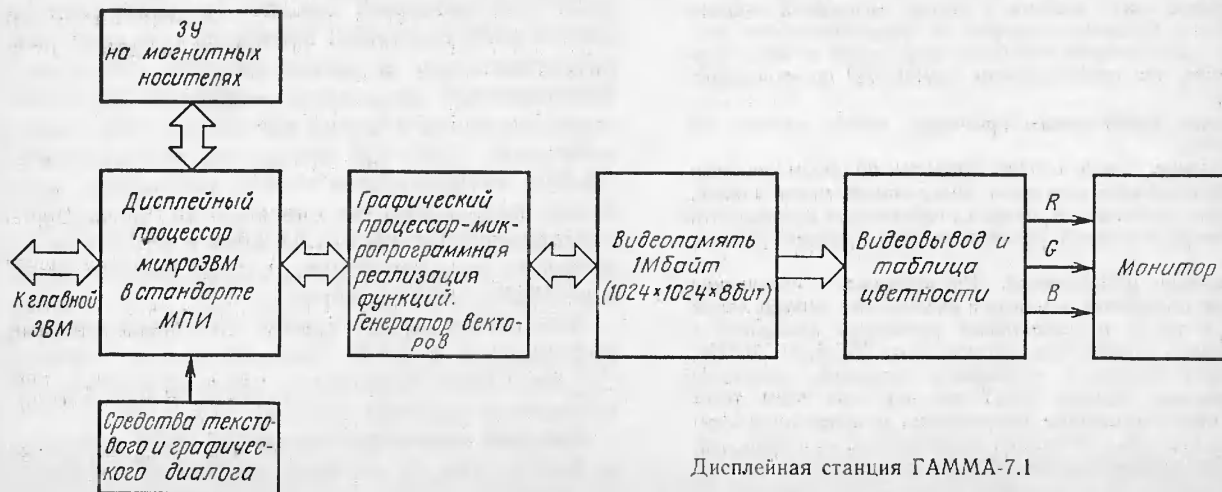
Основное назначение ГП — скоростное функционально-растровое преобразование графических примитивов и генерация пикселей в видеопамати. Высокие требования по скорости работы ГП предопределили его построение на бит-секционном комплекте с реализацией функций на микропрограммном уровне. В состав ГП входит аппаратный генератор векторов, реализующий алгоритм построения по методу цифрового дифференциального анализатора. Конвейеризация этапов обработки высокоуровневого представления графических данных в цепи «Дисплейный процессор — графический процессор — генератор векторов — видеопамать» позволяет приблизиться к реализации идеи «трансфокатора» графической базы данных.

Состав и структура ГП, организация доступа к видеопамати ориентированы на эффективное выполнение следующих функций:

- локального, полного и послойного стирания (раскраски) плоскостей видеопамати;
- формирования и перемещения курсоров различной формы на одной (всех) плоскости видеопамати;
- перемещения массивов пикселей из одного места видеопамати в другое и из плоскости в плоскость;
- выполнения логических и арифметических операций над массивами пикселей;
- пересылки растровых массивов из оперативной памяти ДП в видеопамать и обратно;
- управления аппаратными масштабами и перемещением изображения;
- управления таблицей цветности;
- генерации символов.

Графический процессор, являясь сопроцессором ДП, исполняет дисплейную программу, извлекаемую им по каналу прямого доступа из оперативной памяти ДП. Взаимодействие ГП — ДП основано на безадресном методе (через «почтовую» ячейку). О завершении исполнения дисплейной программы ГП сообщает ДП через организацию прерывания. Реализован ГП на микропроцессорном комплекте БИС серии K1804 и характеризуется следующими данными: объем памяти микропрограмм — 2 Кслов; длина микропрограммного слова — 56 разрядов; разрядность обрабатываемых данных — 16; емкость ПЗУ символов и констант — 2 Кбайт; емкость ОЗУ — 2 Кбайт; длительность такта — 160 или 200 нс (зависит от исполняемой микрокоманды).

Особенностями структуры ГП являются: использование поля адреса ветвления и поля дешифратора кода условия в качестве поля 16-разрядных констант; двунаправленная



Дисплейная станция ГАММА-7.1

внутренняя 16-разрядная магистраль данных; возможности засылки адресов перехода из блока обработки в блок микропрограммного управления для ускорения работы микропрограммных переключателей и организации регистровых пересылок данных, минуя блок обработки.

В дисплейных системах высокого разрешения видеопамять — это ключевой элемент, определяющий качество изображения на экране монитора. В дисплейной станции ГАММА-7.1 видеопамять имеет объем 1 Мбайт и организована в виде двух банков по 512 Кбайт. Банки запускаются по очереди для видеовывода и обмена с графическим процессором: пока один банк находится в режиме обмена, из другого за это время (320 нс) считываются восемь 16-разрядных слов (16 точек растровой строки); затем банки меняются ролями. Необходимый темп видеовывода (50 Мбайт/с) достигается организацией страничного режима считывания в микросхемах памяти. В составе модуля видеопамати имеется 8-разрядный регистр маски, позволяющий закрывать по записи отдельные разряды данных (слои изображения), чем достигается адресация по записи с точностью 1 бит. Это сокращает микропрограммный код и позволяет достичь максимальной скорости генерации графических элементов. Следует подчеркнуть, что процесс обмена с графическим процессором не нарушает работы видеовывода. Блок видеовывода формирует набор необходимых управляющих сигналов, импульсы строчной и кадровой синхронизации для монитора и адреса по чтению для видеопамати, реализует функции аппаратного масштабирования (увеличения изображения в 2 и 4 раза) и перемещения изображения.

Таблица цветности (ТЦ) служит для интерпретации 8-разрядных кодов отображаемых точек изображения в цветостые и яркостные характеристики. ТЦ организована как сверхоперативное запоминающее устройство объемом 256×3 16-разрядных слов и циклом чтения 40 нс. Такая организация позволяет отображать и растровые массивы 16-разрядных данных. ТЦ доступна по записи из ГП во время строчных гасящих интервалов.

Станция выполнена в конструктиве дисплейного процессора, дополненного тремя оригинальными модулями: графического процессора, видеопамати и таблицы цветности. Монитор — высокого разрешения цветного или черно-белого изображения (могут быть использованы также телевизионные мониторы в режимах прогрессивной или черестрочной развертки).

Программное обеспечение ДС состоит из комплекса подпрограмм для ДП и микропрограммы, находящейся в РПЗУ графического процессора. Программы ДП обеспечивают синхронизацию работы дисплейного и графического процессоров, формирование необходимых протокольных команд для графического процессора (структура команд протокола выбрана максимально простой, с тем чтобы ГП затрачивал минимальное время и место для распознавания команд), поддерживают работу графических устройств ввода, которые могут входить в состав дисплейной системы (трекболла буквенно-цифровая и функциональная клавиатура), обеспечивая обработку прерываний от них и формирование, при необходимости, команд ГП по эко-отображению.

Команды графического протокола можно разбить на два класса:

управление. Сюда входят команды по формированию элементов таблицы цветности, содержимого маски записи, масштаба изображения, начала отображения видеопамати на экранную плоскость, реализация полиэкранного режима и т. п.;

построение изображений. Реализованы следующие команды: построение ломаных с различными типами линий (всего 4 типа) и различными режимами занесения в видеопамать (замещение, исключющее ИЛИ, И, ИЛИ); построение текста с размерами символов, кратными «единичному» размеру (5×7 дискрет, при этом текст может иметь различные направления и ориентации символов — всего 16 комбинаций, а также один из перечисленных выше режимов занесения в видеопамать); заполнение

прямоугольных областей в видеопамати; запись строк/столбцов в видеопамать из памяти ДП; чтение строк/столбцов из видеопамати в память ДП и т. п.

Графический процессор (в свободное время занятый тестированием видеопамати), получая задание от ДП, начинает обработку команд графического протокола. О конце работы сообщает ДП формированием прерывания.

Телефон 32-18-50, Новосибирск

Статья поступила 5.05.88

УДК 681.327.69'22—535.6

С. Т. Васьков, О. А. Коломеев, О. В. Перекрест, Г. Ф. Ситников, С. Е. Ткач

ЦВЕТНОЙ МОНИТОР С ВЫСОКИМ РАЗРЕШЕНИЕМ ДЛЯ РАСТРОВЫХ ДИСПЛЕЕВ

В последнее время для автоматизации проектирования применяются специализированные графические станции, использующие для отображения информации мониторы на электронно-лучевые трубки (ЭЛТ). Существуют три основных типа мониторов: на основе запоминающей ЭЛТ, векторный с регенерацией и растровый.

Векторные мониторы с регенерацией стали применять первыми. Однако они имеют высокую стоимость, недостаточные возможности по цветопередаче не обеспечивают сплошное закрашивание поверхностей.

Основные недостатки мониторов на запоминающей ЭЛТ — неспособность к избирательному обновлению части изображения на экране, невозможность отображения быстро меняющейся информации.

Появление быстродействующих полупроводниковых ЗУ с большим объемом запоминаемой информации определило большой интерес к растровым графическим дисплеям. Растровая техника — единственный метод построения дисплея, позволяющий вводить в изображения цвет в виде сплошных площадей с переменной интенсивностью и меняющимися оттенками. Единственный недостаток — эффект ступенчатости, особенно у систем растрового типа с разрешением 256×256 элементов отображения. Эффект ступенчатости можно уменьшить, увеличив разрешающую способность. С появлением микросхем памяти 64 Кбит и более можно выводить на экран монитора изображение объемом 1280×1024 элемента.

Для отображения такого большого объема информации в СКБ вычислительной техники СО АН СССР разработан цветной высокоразрешающий монитор (рис. 1) [1].

Цветная высокоразрешающая ЭЛТ с диагональю экрана 51 см выпускается промышлен-

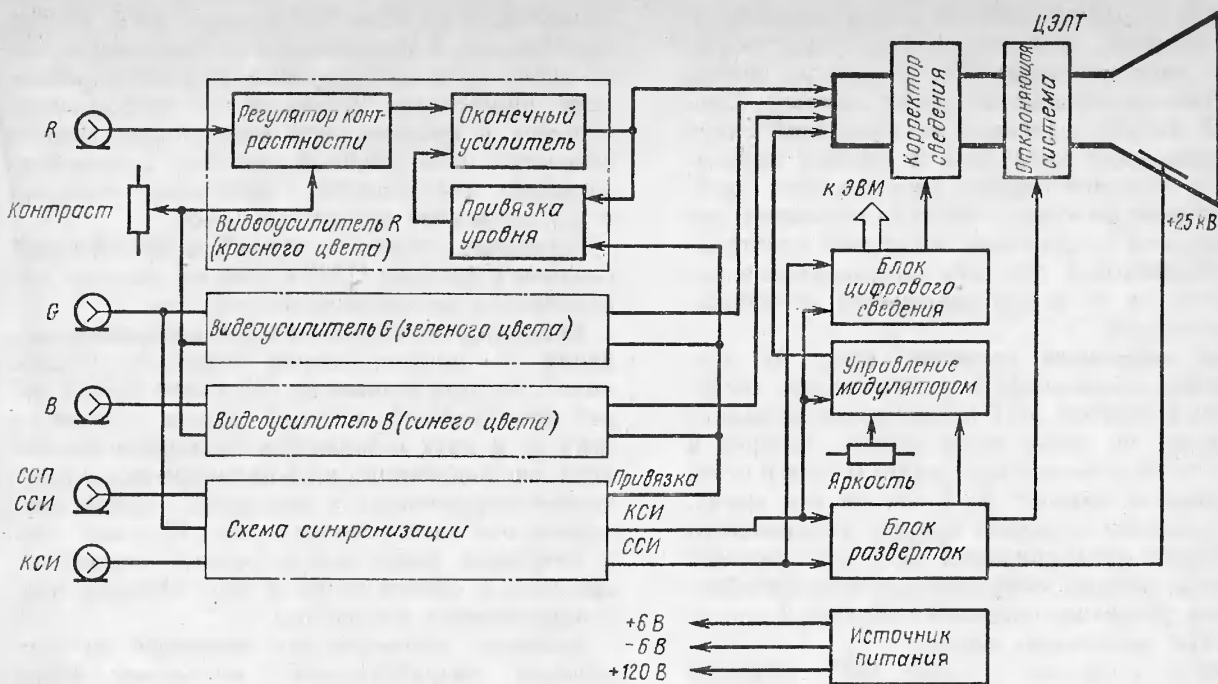


Рис. 1. Структурная схема монитора

ностью в комплекте с отклоняющей системой. Она имеет разрешение в центре по вертикальному клину 1200 линий, по горизонтальному — 1000 линий, антибликовое покрытие экрана, компланарное расположение электронных прожекторов с самосведением лучей, яркость свечения экрана в белом цвете — 50 кд/м².

Особенность блока разверток — работа генераторов кадровой (40...70 Гц) и строчной (28...40 кГц) разверток на повышенных частотах для обеспечения прогрессивной (построчной) развертки раstra и исключения мельканий изображения. Генератор кадровой развертки обеспечивает формирование отклоняющего тока в кадровых отклоняющих катушках ОС, коррекцию линейных искажений, стабилизацию и центровку раstra при изменении кадровой частоты. Генератор строчной развертки вырабатывает ток для строчных катушек ОС, а также напряжение питания ускоряющего электрода цветной ЭЛТ. Он обеспечивает регулировку амплитуды, фазы и линейности отклоняющего тока, а также центровку и коррекцию подушкообразных искажений.

Схема синхронизации формирует из входных сигналов импульсы для блока разверток, вырабатывает импульс привязки уровня черного для усилителей сигналов изображения и импульс гашения обратного хода по кадру. Входными сигналами (рис. 2) являются: отдельные синхронизирующие импульсы строк и кадров (ССИ, КСИ), синхросмесь (ССП) или ви-

деосигнал зеленого цвета с замешанным в него сигналом ССП. Режимы синхронизации переключаются автоматически.

Усилители сигналов R, G и B усиливают входные видеосигналы до 50 В в полосе частот до 45 МГц и позволяют регулировать контрастность, балансировать белый цвет, производить привязку и баланс уровня черного. Все три усилителя одинаковы. Входной видеосигнал поступает через высокочастотный разъем на вход дифференциального усилителя. За счет измене-

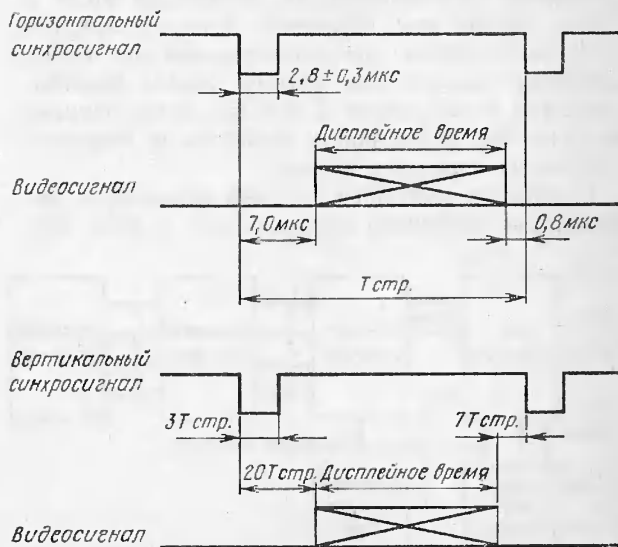


Рис. 2. Временные диаграммы синхросигналов

ния коэффициента усиления дифференциального усилителя регулируется контрастность. После дифференциального усилителя сигнал поступает на вход оконечного усилителя. Оконечный каскад выполнен по каскодной схеме с динамической нагрузкой и охвачен отрицательной обратной связью. Оригинальное схемное решение выходного каскада позволяет для компенсации напряжения запирающего электронных прожекторов смещать модулирующее напряжение на 40 В без увеличения потребляемой мощности.

Блок цифрового сведения (рис. 3) осуществляет статическое и динамическое сведение лучей цветной ЭЛТ и полностью устраняет несведение по всему полю экрана, которое в ЭЛТ с самосведением составляет 0,3 мм в центре экрана и доходит до 1 мм на его края. Блок сведения содержит привод, управляемый импульсами синхронизации, ОЗУ для хранения поправок, каналы цифро-аналогового преобразования, усилители сигналов поправок и многополюсную магнитную линзу.

Память поправок — это ОЗУ объемом 4 Кбайт, выполненное на КМОП-микросхемах. Память позволяет сохранять информацию о поправках с помощью аккумуляторной батареи при отключении монитора и формировать матрицу поправок максимальным размером 16×32 зоны. Типовое число зон 8×16. Разрядность памяти 8 бит, из них 7 бит используется для формирования поправок, а младший бит определяет положение маркера в схеме формирования тестового сигнала.

Привод памяти поправок вырабатывает адреса для памяти поправок и генератора тестового сигнала синхронно с телевизионными развертками. Синхронизация осуществляется стандартными телевизионными сигналами КСИ и ССИ. Число зон поправок программируется переключателями, расположенными на плате. Опорная частота для работы схемы вырабатывается генератором с ФАПЧ, позволяющим модулю без перестройки работать в широком диапазоне строчных частот.

Генератор тестового сигнала формирует видеосигнал тестового изображения в виде сет-



Рис. 3. Схема блока сведения

чатого поля 8×8 зон. По данному тесту можно производить и контролировать сведение лучей по всему полю изображения. Генератор позволяет программно формировать изображение маркера в произвольной зоне экрана в виде сетчатого поля двойной частоты. Положение маркера определяется наличием единицы в младшем бите памяти поправок.

Выходные усилители мощности преобразуют сигналы с выходов ЦАП в токи отклонения для корректора динамического сведения.

Корректор статического и динамического сведения — многополюсная электромагнитная линза, расположенная на горловине ЦЭЛТ перед отклоняющей системой. Линза состоит из двух 4- и двух 6-полюсных электромагнитных линз, расположенных на 8-цилиндрических магнитных сердечниках в плоскости, перпендикулярной оси электронного пучка. Изменяя токи в катушках линз, можно менять положение красного и синего лучей в двух взаимно перпендикулярных плоскостях.

Значения динамических поправок по специально разработанному алгоритму определяются в ЭВМ и через блок сопряжения с магистралью записываются в ОЗУ поправок. После коррекции сведения лучей ЭВМ отключаются от блока поправок. Для сохранения поправок в ОЗУ при включении предусмотрен резервный аккумулятор.

Основные технические данные монитора

Размеры рабочего поля экрана ЭЛТ, мм	280×372
Частота строчной развертки, кГц	28...40
Частота кадровой развертки Гц	40...80
Полоса видеотракта, МГц	до 45
Число адресуемых точек при F=50 Гц:	
прогрессивная развертка, элементов	1024×768
чересстрочная развертка, элементов	1280×1024
Несведение лучей, мм	Не менее 0,3 по всему полю экрана

Монитор предназначен для работы в составе дисплейной станции высокого разрешения. Примером такой станции может служить цветная полутоновая дисплейная система профессионального назначения Гамма-7.1 [2].

Телефон 35-33-61, Новосибирск

ЛИТЕРАТУРА

1. Васьков С. Т., Коломеев О. А. и др. Цветной монитор с высоким разрешением для растровых дисплеев // Тез. докл. на IV Всесоюз. конф. по проблемам машинной графики. — Серпухов, 1987.
2. Бучнев А. А., Сизых В. Г. Цветная полутоновая дисплейная система профессионального назначения Гамма-7.1 // Тез. докл. на IV Всесоюз. конф. по проблемам машинной графики. — Серпухов, 1987.

Статья поступила 21.03.88

В. Ю. Липерь, В. Ф. Тарасевич, А. А. Горуневич

ЦВЕТНОЙ СТРУЙНЫЙ ГРАФОПОСТРОИТЕЛЬ

Для изготовления «твердых копий» экранного изображения и полноформатных бумажных документов в цвете (рисунков многослойных плат и БИС, снимков, обработанных на ЭВМ, и др.) необходимы цветные регистрирующие устройства, совместимые по форматам графических данных с дисплейными рабочими станциями. Традиционные координатографы со сменными пишущими узлами оказываются малоэффективными или вовсе неприменимыми при выводе изображений, содержащих большое число линий и областей сложной конфигурации или передаваемых из видеопамати дисплея в виде растровой матрицы кодов цвета. Для этих целей наиболее целесообразно использовать растровые регистрирующие устройства в сочетании с электростатической или струйной печатью. Из двух последних способов регистрации лучшими перспективами по разрешающей способности и качеству цветопроизведения обладает струйный способ, обеспечивающий возможность получения производных цветов за счет наложения друг на друга прозрачных тонеров основных цветов.

Разработанный в Институте технической кибернетики АН БССР цветной растровый струйный графопостроитель включает двухкоординатный механизм с вращающимся барабаном, перемещаемую с помощью винтовой передачи каретку, на которой установлена 4-секционная голов-

ка струйной записи и узлы рециркуляции чернил. Характеристики графопостроителя приведены ниже.

Формат рабочего поля, мм	594×841 (A1)
Шаг дискретизации, мм	0,2
Максимальная частота модуляции капель, кГц	50
Погрешность регистрации растр-элементов, мм	0,1
Число основных цветов	4

Цветное изображение формируется из четырех основных тонов (черного, желтого, голубого, пурпурного) субтрактивным синтезом при прямом наложении тонов и пространственно-аддитивным синтезом растр-элементов.

Использование метода непрерывной генерации красящих капель позволяет достичь на порядок более высоких скоростей записи, чем при генерации капель «по требованию». Поток капель модулируется путем заряда и отклонения в электрическом поле капель, подлежащих нанесению на бумагу. Неиспользованные капли возвращаются в систему рециркуляции чернил.

Процесс растровой записи управляется с помощью встроенного контроллера, подключенного к шине ЭВМ или выходу специального процессора «вектор-растр», преобразующего графические данные из массива графических примитивов стандарта GKS (линия из отрезков, область заполнения, символ и др.) в последовательность упорядоченных по строкам кодов цвета растр-элементов изображения.

220072, Минск, 72, Сурганова, 6, НТК АН БССР;
тел. 39-50-78

Сообщение поступило 11.02.88

ЯП ПАСКАЛЬ — ВОПРОСЫ РЕАЛИЗАЦИИ

С. Ф. Добриневский

ПРОЦЕДУРЫ ОБРАБОТКИ ПЕРЕРЫВАНИЯ В ПАСКАЛЕ ОС ДВК

Отсутствие в стандартном Паскале и в его широко используемой реализации для компьютеров, работающих под управлением ОС ДВК (РАФОС) [1], процедур обработки прерываний значительно ограничивает применимость этого языка в системах автоматизации научного эксперимента, управляющих системах реального времени. Имеющиеся решения этой проблемы (например, [23]) отличаются ориентацией на определенный круг задач, в то время как необходимы соответствующие программные средства общего назначения.

Пакет INPROC предназначен для организации обработки прерывания исключительно средствами Паскаля и включает в себя три процедуры. Пакет объемом 3 блока на гибком магнитном диске может поставляться в виде объектного модуля, аналогично пакету прямого доступа к элементам файловой структуры RANIO, входящему в состав системы программирования Паскаль для ДВК. При использовании процедур обработки прерываний они должны быть объявлены программистом как внешние в соответствии с синтаксисом Паскаля ОСДВК и включены в загрузочный модуль задачи при компоновке. Никакой специальной

инициализации для работы с пакетом не требуется, так как связи между векторами прерываний и пользовательскими процедурами устанавливаются и отменяются в процессе работы с помощью процедур, входящих в состав пакета.

В описаниях процедур параметр VECTOR принимает значения, равные адресам векторов прерывания, параметр PRIORITY (приоритет процессора) — значения от 0 до 7 в соответствии с архитектурой ЭВМ линии ДВК. Процедура обработки прерываний NEWPROC может быть произвольной и должна располагаться в разделе описаний той процедуры, в которой используется обработка прерываний, или объемлющей процедуры. Поскольку процедура NEWPROC является резидентной в оперативной памяти компьютера (это требование вполне естественно), она не должна находиться в оверлейных сегментах программы. На сам пакет INPROC ограничения, связанные с оверлейностью, не распространяются.

1. Присоединение процедуры к вектору прерывания:

```
procedure CONNECT (VECTOR, PRIORITY: integer,
                    procedure NEWPROC)
```

Процедура присоединения CONNECT позволяет устанавливать связи между векторами прерываний и пользовательскими процедурами в процессе работы программы. Для установ-

ления таких связей при каждом обращении к CONNECT путем динамического размещения генерируется тело первичной процедуры обработки прерываний объемом около 100 байт.

После того, как выполнено присоединение, при возникновении прерывания по вектору VECTOR управление передается первичной процедуре обработки прерываний, а затем, если приоритет процессора PRIORITY, указанный при вызове CONNECT, не меньше текущего приоритета процессора, — на начало процедуры NEWPROC. Сравнение приоритетов выполняется программным путем, однако следует понимать, что в ЭВМ «Электроника 60» и совместимых с ней старший бит приоритета аппаратным образом используется для запрещения/разрешения внешних прерываний. Поэтому, хотя приоритет процессора «Электроника 60» может принимать значения 0...7, при установленном старшем бите (приоритет 4 и выше) прерывания всегда будут запрещены. Например, при текущем приоритете процессора 2 управление может быть передано по прерыванию процедуре, присоединенной с помощью CONNECT с приоритетом 3, однако при текущем приоритете, равном 5, прерывания не произойдет и управление не может быть передано. Возврат из прерывания производится после завершения процедуры NEWPROC и не требует каких-либо специальных действий.

Присоединение NEWPROC к вектору прерывания, хотя и отменяет действие прежней процедуры обработки прерывания (до первого использования CONNECT это обработка, выполняемая ОС), не исключает возможности воспользоваться этой процедурой.

2. Использование отмененной процедуры обработки прерывания:

```
procedure CONTINUE (VECTOR: integer)
```

Процедура продолжения CONTINUE используется внутри пользовательских процедур обработки прерываний и означает передачу управления процедуре, связанной с данным вектором прерывания до выполнения CONNECT. После завершения этой процедуры возврат из прерывания не происходит, а управление передается на оператор, следующий за вызовом CONTINUE. Таким образом, возможно не заменять, а дополнять вновь присоединенной процедурой NEWPROC обработку прерываний по любому вектору. В частности, CONTINUE позволяет использовать наряду с пользовательской системной обработкой прерываний, например от клавиатуры системного терминала.

Если к вектору прерывания были последовательно присоединены несколько процедур (и, соответственно, несколько процедур после-

довательно отменены), то действие CONTINUE сведется к выполнению только последней отмененной процедуры.

3. Восстановление исходного вектора:

```
procedure RESTORE (VECTOR: integer)
```

Действие этой процедуры обратно действию процедуры присоединения CONNECT. К указанному вектору присоединяется прежняя процедура обработки прерываний, а тело процедуры первичной обработки уничтожается с освобождением соответствующего объема памяти компьютера. Если к одному и тому же вектору последовательно были присоединены несколько различных процедур обработки прерываний, то последовательное применение RESTORE будет приводить к восстановлению связи вектора прерывания с этими процедурами в порядке, обратном порядку присоединения, вплоть до восстановления системной обработки прерываний. Физически это означает, что каждая первичная процедура обработки прерываний, созданная при работе CONNECT, содержит ссылку на отмененную прежнюю процедуру, и, следовательно, при последовательном присоединении к одному и тому же вектору процедуры обработки прерываний образуют связанный список.

Рассмотрим простейшие способы использования процедур обработки прерываний. Работа с прерываниями теперь уже возможна без обращения к ассемблеру, хотя от программиста и требуется владение некоторыми элементарными представлениями об архитектуре компьютера. В первую очередь, это относится к количеству и адресам векторов прерываний. В рассматриваемых здесь примерах упоминаются только два типа внешних прерываний — от клавиатуры с вектором 60В и от таймера с вектором 100В (в Паскале ОС ДВК символ В означает восьмеричное число).

Общая структура программы, использующей пакет INPROC, примерно такова:

```
program TEST (input, output);  
  
const  
    KBDVEC = 60В; {вектор прерывания от клавиатуры}  
  
    TIMVEC = 100В; {вектор прерывания от таймера}  
  
var  
    OVER: boolean;  
  
procedure STRIKE;  
  
begin  
    {обработка прерываний от клавиатуры}  
  
end;  
  
procedure TIMING;
```



```

begin
  {обработка прерываний от таймера}
end;
procedure CONNECT (VECTOR, PRIORITY: integer,
  procedure NEWPROC); external;
procedure RESTORE (VECTOR: integer); external;
begin
  OVER := false;
  CONNECT (KBDVEC, 3, STRIKE);
  CONNECT (TIMVEC, 2, TIMING);
repeat
  {основная часть программы повторяется,
  пока не будет выставлен флаг завершения}
until OVER;
{завершение выполнения}
RESTORE (KBDVEC);
RESTORE (TIMVEC)
end.

```

В приведенном примере к векторам прерываний от клавиатуры и от таймера присоединяются пользовательские процедуры с приоритетами соответственно 3 и 2. Это означает, что обработка прерывания от клавиатуры будет происходить даже в том случае, когда в данный момент обрабатывается прерывание по вектору 100В. В свою очередь, выполнение обеих процедур — STRIKE и TIMING — может быть прервано, например при ошибке обращения к каналу.

Процедуры обработки прерываний могут быть в принципе произвольными, однако в каждом конкретном случае существуют ограничения, связанные с параллельным характером выполнения всей программы в целом. Так, может оказаться невозможным использование в процедурах обработки прерываний локальных переменных. Это связано с тем, что при повторном входе в процедуру обработки прерывания значения локальных переменных не сохраняются, поэтому их имеет смысл использовать только в пределах одного вхождения.

Возможны такие ситуации, когда выполнение процедуры обработки прерывания по какому-либо вектору может быть приостановлено в результате возникновения прерывания по тому же вектору (вложенные прерывания). В этом случае процедура обработки прерывания вообще не должна иметь локальных переменных: Глобальные переменные следует применять и для сохранения некоторой информации в течение промежутка времени между прерываниями. Например, «часы» могут быть реализованы следующим образом:

```

procedure TIMING;
begin
  TICK := succ (TICK); {подсчет числа прерываний от
  таймера (частота 50 Гц)}
  if TICK > 50 then begin
    {через каждые пятьдесят прерываний сброс
    переменной TICK, увеличение счетчика секунд на
    единицу, каждые 60 секунд переход к новой
    минуте и т.д., вывод показаний "часов"
    на дисплей в определенной позиции экрана}
  end
end;

```

Для этой процедуры обработки прерываний принципиально важно сохранение значений счетчиков тиков, секунд, минут и часов в течение всего времени выполнения программы TEST. Поэтому соответствующие переменные должны быть декларированы в разделе описаний головной программы (и, разумеется, определены до выполнения присоединения). Однако «часы» можно осуществить и с использованием системной поддержки сетевого таймера:

```

procedure TIMING;
  var CURRENTTIME: real;
  procedure CONTINUE (VECTOR: integer); external;
begin
  CONTINUE (TIMVEC);
  TICK := succ (TICK);
  if TICK > 50 then begin
    CURRENTTIME := TIME;
    {сброс счетчика тиков, вычисление показаний
    часов, вывод показаний на дисплей}
  end
end;

```

В этом варианте процедуры TIMING использование CONTINUE приводит к тому, что при каждом прерывании наращивание счетчика тиков производится средствами ОС, поэтому обращение к определенной в данной реализации Паскаля функции TIME дает правильный результат. Как глобальная переменная описывается только пользовательский счетчик тиков, необходимый для того, чтобы не обновлять показания на дисплее слишком часто. Использование функции TIME в предыдущем примере давало бы постоянно один и тот же результат, соответствующий показанию системного таймера до выполнения процедуры присоединения.

Обработка прерываний от клавиатуры чаще всего применяется для реализации различных

ПРОГРАММИРОВАНИЕ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ НА ЯЗЫКЕ ПАСКАЛЬ

Наблюдается устойчивая тенденция к широкому применению языков высокого уровня для программирования систем реального времени. Создаются специализированные языки реального времени (Modula, Ada, Quasic) [1, 2].

На микроЭВМ типа «Электроника 60» и ДВК широко применяется язык Паскаль [3]. Основное препятствие для его использования в системах реального времени — невозможность работы Паскаль-программ на автономной ЭВМ без поддержки стандартной операционной системы (ОС ДВК); отсутствие встроенных средств программирования параллельных процессов (обработки прерываний) и механизма программной обработки ошибок, возникающих при выполнении. Однако такие особенности компилятора Паскаля для ДВК, как возможность размещения переменных по абсолютным адресам памяти, включения в текст программы ассемблерных вставок и отдельной компиляции модулей, позволяют обойти эти препятствия.

С использованием указанных особенностей компилятора была разработана система программирования на Паскале для реального времени. Она включает в себя следующие компоненты:

1. **Работа Паскаль-программ без поддержки ОС.** Чтобы отказаться от поддержки ОС, было решено пойти по пути написания ее эмуляторов. Такой подход более гибок, чем изменение кодов в библиотеке Паскаля, и позволяет легко переносить программы на ЭВМ с различной аппаратной конфигурацией.

Основной канал связи исполнительной системы Паскаля с операционной системой ОС ДВК (RAFOS) — программные EMT-запросы. Эмулирующий ОС модуль должен обеспечивать выполнение по крайней мере следующих из них: установки максимально доступной памяти, ввод и вывод с терминала, окончание работы программы. С точки зрения пользователя, такой эмулятор представляет собой небольшой объектный модуль, который подсоединяется программе на стадии компоновки (LINK).

Модулей, эмулирующих ОС, написано несколько, для различных аппаратных конфигураций: ДВК1 с терминалом и без; ДВК1 с нестандартными регистрами терминала, с сокращенной памятью; персональный компьютер БК-0010 [4]. Наиболее полный из них — модуль, обеспечивающий работу программ на ДВК1 с терминалом, размером около 250 слов.

режимов управления выполнением программы. Так, во многих случаях важно иметь возможность прекращения выполнения программы или какой-либо процедуры в произвольный момент времени. Использование предусмотренного в ОС для этой цели кода 003 (cancel, CY/C) вызывает аварийное завершение и потому нежелательно. В приведенном ниже примере тот же код прекращает выполнение программы другим образом:

```
procedure STRIKE;
var
  CH: integer;
  KBCSR origin 177560B,
  KBDAT origin 177562B: integer;
procedure CONTINUE (VECTOR: integer); external;
begin
  CH := KBDAT;
  if CH = 3 then OVER := true
  else CONTINUE (KBDVEC)
end;
```

Процедура CONTINUE, как видно из текста, не выполняется в случае поступления с клавиатуры кода 003, поэтому требование прекращения выполнения не станет известным ОС, а установление флага OVER приведет к завершению выполнения программы по алгоритму, определенному программистом. Использование процедуры продолжения в этом примере дает возможность, несмотря на присоединение пользовательской обработки прерываний, работать по-прежнему с обычными процедурами Паскаля read и readln, предназначенными для ввода с клавиатуры. Иными словами, происходит «перехват» и некоторый предварительный анализ вводимых с клавиатуры символов до того, как они обрабатываются исполняющей системой Паскаля и утилитами ОСДВК.

Эти примеры иллюстрируют лишь некоторые варианты применения пакета INPROC. Процедуры обработки прерываний целесообразно использовать для управления автоматизированным оборудованием, реализации активного диалога, создания непосредственно управляемых систем.

220602, Минск, Ленинский пр., 70, Ин-т физики АН БССР

ЛИТЕРАТУРА

1. Бутомо И. Д., Самочадин А. В., Усанова Д. В. Программирование на алгоритмическом языке Паскаль для микроЭВМ. — Л.: Изд-во ЛГУ, 1986. — 214 с.
2. Журавлев В. И., Сапожников С. В., Шинкевич С. Л. Процедуры работы с аппаратурой КАМАК на языке Паскаль // В кн.: Проблемно-ориентированные измерительно-вычислительные комплексы. — Новосибирск: Наука, 1986. — С. 62—65.

Статья поступила 11.05.88

Он состоит из блока инициализации, который выполняется непосредственно после запуска программы, а затем передает управление исполнительной системе Паскаля, и системы обработки программных запросов. Основную часть его составляет поддержка ввода-вывода с терминала. Она выполнена так, что пользователь (и программа) не отличает ее от системы ОС ДВК. Действуют все те же функциональные клавиши прерывания/продолжения вывода на терминал, прерывания выполнения программы и т. п.

Дополнительно реализована функция запросов, аналогичная оператору REQUEST языка Quasic-2. Она позволяет задавать клавишу, при нажатии на которую выполнение программы прерывается и управление передается на определенную пользователем процедуру. Такое соответствие устанавливается процедурой REQUEST с видом описания

```
procedure REQUEST (KEY: char; procedure PROC);  
external;
```

где KEY — символ; при нажатии на клавишу, вырабатывающую код данного символа, будет выполняться процедура PROC.

Нулевое значение KEY(CHR(0)) отменяет действие функции запросов. При отработке нажатия на запрограммированную таким образом клавишу указатель стека поднимается до верхнего своего положения, обеспечивается доступ к глобальным переменным (устанавливается регистр R5) и управление передается процедуре, заданной пользователем. Эта процедура должна содержать внутри себя или переход на метку, определенную в основной программе, или бесконечный цикл (монитор) и не выполняться до конца. В процедуре можно использовать любые глобальные и локальные переменные. В каждый момент времени запрограммированной может быть только одна клавиша. Однако в процессе выполнения как клавиши, так и процедуры их отработки можно менять сколько угодно раз.

Функция запросов реализована двумя способами: «жестким» и «мягким». В случае «жесткого» REQUESTа стек поднимается непосредственно после нажатия клавиши, но это может произойти внутри процесса или вложенных процессов обработки прерываний, что, естественно, нарушает их работу. При «мягком» REQUESTе указатель стека поднимается только после окончания всех процессов обработки прерываний, на нулевом уровне вложенности, и не влияет на их работу. Такая «мягкость» обеспечивается для прерываний, обрабатываемых посредством SETVEC (см. ниже).

2. Обработка прерываний на Паскале. Обработка прерываний организуется с помощью процедуры SETVEC, написанной в основном

на ассемблере. Используя ее, можно любому вектору прерывания поставить в соответствие процедуру, написанную на Паскале, так, что каждый раз при возникновении прерывания по данному вектору, будет выполняться процедура SETVEC, имеющая описание

```
procedure SETVEC (VEC, PSW: integer; procedure  
PROC); external;
```

где VEC — вектор прерывания, в соответствие которому ставится процедура; PSW — приоритет прерывания, это число записывается во второе слово вектора прерывания и, соответственно, в слово состояния процессора при возникновении прерывания; PROC — процедура (процесс) обработки прерывания.

Процедура обработки прерывания не должна содержать передаваемых ей параметров и переходов на метки, определенные в других процедурах. На использование глобальных или локальных переменных не накладывается никаких ограничений. Прерывания обрабатываются по следующему алгоритму: сохраняются регистры общего назначения, обеспечивается доступ к глобальным переменным, управление передается на процедуру, определенную пользователем, после ее выполнения регистры восстанавливаются и производится возврат из прерывания. Вся система, обеспечивающая обработку прерываний, содержится в SETVEC и занимает около 60 слов. Вектор прерывания должен иметь значение менее 400₈. Одновременно можно обрабатывать прерывания от 64 различных векторов, причем допустима их переустановка в процессе выполнения сколько угодно раз. От момента возникновения запроса на прерывание до начала выполнения процедуры пользователя на процессоре с быстродействием около 250 тыс. команд в секунду проходит около 100 мкс.

3. Обработка ошибок. Некоторые версии Паскаля не имеют встроенного механизма обработки ошибок, возникающих при выполнении программы, или имеют механизм, не удовлетворяющий требованиям системы реального времени. Для адаптивной обработки ошибок выполнения были внесены минимальные изменения в библиотеку Паскаля — PASCAL.OBJ. Теперь, с помощью процедуры SETERR, можно заменить стандартную обработку ошибки (печать сообщения на терминале и переход в монитор ОС) на выполнение процедуры, установленной пользователем. Описание процедуры SETERR имеет вид

```
procedure SETERR (ERR: integer; procedure PROC);  
external;
```

где ERR — номер ошибки (всего около 20 типов ошибок выполнения); PROC — процедура, ко-

торая будет выполняться при возникновении ошибки с данным номером. Для такой процедуры действуют те же возможности и ограничения, что и для SETVEC.

4. Программирование аппаратуры КАМАК.

Одно из широко используемых устройств сопряжения ЭВМ с экспериментальной установкой — аппаратура в стандарте КАМАК. Поэтому были разработаны специальные средства программирования такой аппаратуры.

Решено отказаться от создания специальных процедур для работы с КАМАК из-за того, что накладные расходы в этом случае могут намного превышать полезное действие. Программирование КАМАК посредством обращения по абсолютным адресам памяти также неудобно, так как требует или их ручного вычисления, или вычисления всех адресов на стадии выполнения, что не эффективно. Кроме того, в ряде приложений требуется максимально быстрая работа с КАМАК. Поэтому пошли по пути создания специальных макроопределений, которые включались бы в текст программы как ассемблерные вставки. Такой подход позволяет создавать макробιβотеки, «подцеплять» их на стадии MACRO, вычислять все адреса на этапе компиляции и генерировать оптимальный рабочий код.

Реализованы следующие макроопределения для работы с КАМАК: ZERO, ACT, QUERY. Они по форме аналогичны операторам языка Quasic-2. Макроопределения имеют вид

```
ZERO CRATE
ACT FUNC, CRATE, MOD, SUBADR, D1, D2, L
QUERY CRATE, Q, L;
```

где CRATE, FUNC, MOD, SUBADR — соответственно номер крета, выполняемой КАМАК-функции, модуля и субадрес. На их место должны подставляться либо цифры, либо имена констант. Это позволяет вычислять все абсолютные адреса на стадии компиляции. D1, D2 — младшая и старшая части КАМАК-слова, Q — отклик Q, D1, D2, Q должны быть определены в программе на Паскале как целые переменные.

ZERO производит одноименную операцию над заданным крейтом.

ACT выполняет ту или иную КАМАК-функцию. Если это функция чтения ($0 < \text{FUNC} < 7$), то информация из КАМАК записывается в D1 и D2 (если D2 задано). Аналогично, в случае функции записи ($16 < \text{FUNC} < 23$) данные считываются из D1 и D2 в КАМАК. Символ L указывает, что D1 и D2 являются локальными переменными, определенными в той процедуре, откуда вызывается данный ACT. Если пропущены параметры, задаваемые константами, то они берутся из предыдущего ACT. Если пропущена функция

ACT, CRATE..., то она не переустанавливается в регистре состояния контроллера.

При одновременном использовании старшей части КАМАК-слова в процессах обработки прерываний и в основной программе могут возникать ошибки из-за того, что старшая часть КАМАК-слова является общим разделяемым ресурсом, причем процессы обработки прерываний не в состоянии ее сохранять. Поэтому при работе со старшей частью КАМАК-слова дополнительно генерируются коды, обеспечивающие правильное ее использование путем повторного выполнения команды, если она была нарушена прерыванием. В зависимости от функции макроопределение ACT генерирует код, занимающий 2...20 слов.

QUERY позволяет получить в программе отклик Q. Однако (как показывает практика) удобнее работать с откликом Q, непосредственно обращаясь к регистру состояния контроллера по абсолютному адресу. То же можно сказать о регистре маски и запроса. Необходимо иметь в виду, что описанные выше макроопределения рассчитаны на контроллеры типа K-16.

Перед использованием этих макроопределений должно быть вызвано макроопределение PASMAL, определяющее другие макроопределения. В программе DIAG макроопределения применяются для построения на экране графического дисплея красной диагональной линии с помощью модуля График-цвет:

```
program DIAG;
const CRATE=1; GRAF=2; SUBADR=15;
var I,D1,D2, CSR origin 162000B: integer;
(*RC .Mcall PASMAL
PASMAL *)
begin
(*RC ZERO CRATE *)
D1:=0; D2:=2;
(*RC ACT 16,,GRAF,SUBADR,D1,D2 *)
for I:=1 to 255 do begin
D1:=I+I*256;
(*RC ACT ,,,D1,D2 *)
while CSR >= 0 do;
end;
end.
```

Процедура получения загрузочного модуля этой программы имеет вид

```
RUN PASCAL DIAG.MAC=DIAG.PAS
RUN MACRO DIAG.OBJ=DIAG.MAC,PASCAL.MLB/M
RUN LINK DIAG.SAV=DIAG.OBJ,PASCAL.OBJ,
```

где PASCAL.MLB — макробιβотека работы с КАМАК.

Каждый из четырех указанных компонентов (работа без поддержки ОС, обработка прерываний, обработка ошибок и программирование КАМАК) можно использовать по отдельности. Например, работая под управлением ОС ДВК (RAFOS), можно обрабатывать прерывания и (или) программировать КАМАК. Вся система

вместе с документацией занимает около 350 блоков на диске.

Рассмотренная методика программирования систем реального времени на языке Паскаль широко применяется для создания программной системы управления и контроля разрезного микротрона НИИЯФ МГУ. Все существующее программное обеспечение написано с применением этой методики. Практика показывает большое удобство создания и надежность эксплуатации программ реального времени, подготовленных в рамках такого подхода.

Телефон 939-13-24, Москва.

УДК 681.3.06

А. Г. Горшенин, В. Б. Воронов

ЗВУКОВОЕ СОПРОВОЖДЕНИЕ ПРОГРАММ В ПАСКАЛЕ M86 ДЛЯ ПЭВМ ЕС1840

Руководство пользователя языка Паскаль M86 не содержит процедуры работы со звуковым сопровождением, а процедуры SOUND и NOSOUND, соответствующие версии языка, поставленной в составе ПО ЕС1840, при попытках применения не обрабатываются компилятором. Невозможность звукового сопровождения программ, написанных на языке Паскаль, вызывает и разочарование, и недоумение.

Все же затруднение можно обойти, используя дополнительные возможности Паскаля M86 для получения доступа к портам микроЭВМ.

С помощью фрагмента программы (рис. 1) запрограммировали счетчик-делитель таймера микроЭВМ на деление тактовой частоты генератора (коэффициент деления загружается в таймер младшим (MB) и старшим (SB) байтами). Обращение к порту 61H включает звук громкоговорителя (49H) или выключает его (49H).

Для нот первой и второй октав коэффициенты деления даны в таблице (рис. 2).

При разработке прикладных программ применили два вида процедур: Z и ZV, приведенных на рис. 3, 4.

Процедура Z загружает таймер коэффициентом деления, задаваемым программистом в виде десятичного числа. Начальную установку таймера и разре-

```

program ZVUK;
var
  MB, SB: integer;
begin
  PORT[$43] := $86; {Программирование С42-деление частоты}
  PORT[$61] := $4B; {Включение громкоговорителя}
  PORT[$42] := MB;
  PORT[$42] := SB;
end.

```

Рис. 1. Текст программы ZVUK

```

procedure Z(K: integer);
begin
  PORT[$42] := 10(K);
  PORT[$42] := HI(K);
end;

```

Рис. 3. Текст процедуры Z

шение звучания громкоговорителя микроЭВМ проводят первый и второй операторы (см. рис. 1) в необходимом месте программы. Перед окончанием работы программ, применяющих процедуру Z(K), необходимо подать в порт

1. Янг С. Алгоритмические языки реального времени.— М.: Мир, 1985.
2. Подольский Л. И., Лясковский А. П. Система программирования Quasic-2 для микроЭВМ // Микропроцессорные средства и системы.— 1987.— № 2.— С. 9—11.
3. Йенсен К., Вирт Н. Паскаль.— М.: Финансы и статистика, 1982.
4. Завилов В. Н., Константинов М. Ю., Померанец М. В. Программирование на языке Паскаль для микроЭВМ «Электроника БК-0010» // Микропроцессорные средства и системы.— 1987.— № 1.— С. 37—39.

Статья поступила 16.02.88

Октава	Нота	Коэффициент деления
I	ДО	9180
	ДО#	9011
	РЕ	8179
	РЕ#	7719
	МИ	7285
	ФА	6877
	ФА#	6492
	СОЛЬ	6127
	СОЛЬ#	5783
	ЛЯ	5458
ЛЯ#	5152	
СИ	4864	
II	ДО	4591
	ДО#	4333
	РЕ	4089
	РЕ#	3860
	МИ	3644
	ФА	3438
	ФА#	3245
	СОЛЬ	3064
	СОЛЬ#	2891
	ЛЯ	2729
ЛЯ#	2576	
СИ	2432	

Рис. 2. Таблица коэффициентов

61H байт отключения громкоговорителя. Процедура ZV отключает громкоговоритель после выдачи звукового сигнала длительностью D мс. Таким образом, с помощью этой процедуры программируется звуковой фрагмент,

```

procedure ZV(K, D: integer);
begin
  PORT[$43] := $86; {Программирование делителя}
  PORT[$61] := $4B; {Включение громкоговорителя}
  PORT[$42] := 10(K); {Младший байт коэффициента деления}
  PORT[$42] := HI(K); {Старший байт коэффициента деления}
  DELAY(D); {Длительность звучания в мсек}
  PORT[$61] := $49; {Отключение громкоговорителя}
end;

```

Рис. 4. Текст процедуры ZV

т. е. задаются параметры звука (нота и длительность) — достаточно удобно.

Телефон 52-43-84, Севастополь
Сообщение поступило 2.01.89

РЕКЛАМА

Кооператив «Электрон» предлагает владельцам ПЭВМ: ДВК, БК-0010, РК-86, «Микроша», «Специалист», «Спектр», широкий выбор системных, прикладных, учебных, игровых программ.

Закключаем с авторами договоры на тиражирование разработанного ими программного обеспечения.

Предприятиям, учреждениям, кооперативам оказываем информационное содействие в реализации и приобретении ЭВМ всех типов.

Наш адрес: 103489, Москва, корп. 705, кооператив «Электрон».

Телефон 536-12-81 (по вторникам, четвергам и субботам с 18 до 20 ч.)

А. Г. Водяник

НОВАЯ РЕАЛИЗАЦИЯ ЯЗЫКА ПАСКАЛЬ ДЛЯ РАФОС

Предлагаемая система программирования языка Паскаль работает по технологии прямой слитной компиляции: входные файлы — загрузочный модуль. Она отличается высоким быстродействием компилятора, компактностью, минимальной потребностью во внешней памяти.

Система состоит из однопроходного компилятора P SAV и исполняющей системы PMON.SAV с размерами 68 и 16 блоков. Предусмотрена возможность применения нескольких версий исполняющей системы. Генерируемый компилятором загрузочный модуль содержит обращения к PMOV в виде команд TRAP. Если компиляция завершается успешно, то файл PMON.SAV присоединяется к загрузочному модулю, образуя готовую к выполнению программу. При компиляции не создаются промежуточные файлы, не выполняется редактирование связей, поэтому достигается скорость 10...15 строк/с и возможна комфортная работа с накопителями малой емкости. Хорошие результаты достигаются даже при использовании одного квазидиска емкостью 128 Кбайт.

Компилятор принимает командную строку в формате интерпретатора CSI, формирует листинг и карту загрузки. Листинг содержит параллельно со строками исходных файлов соответствующие адреса кода в загрузочном модуле, что облегчает поиск ошибок времени выполнения.

В реализованной версии языка идентификаторы могут включать наряду с латинскими русские буквы, в том числе Ч, Ш, Щ, Э, Ю, знаки %, Ъ и подчеркивания. Спецификатор адреса ORIGIN можно использовать для указания адреса переменной и установки смещения локальной переменной относительно указателя стека, а также совместно с полем записи для установки смещения поля от начала записи. Во входном файле могут находиться директивы вложения других входных файлов. Имеются ключи для задания проверок стека и индексов массивов, управления листингом, а также средства для вставки в программу фрагментов в машинном коде.

Компилятор P написан на языке Паскаль, размер исходного текста — 3700 строк. Исполняющая система скомпонована с использованием модулей библиотеки PSLIB OMSI-PASCAL1. Система программирования может быть адаптирована к микроЭВМ, подобным БК-0010.

341004, Донецкая область, Мариуполь,
ул. Заозерная, 98-Б, Водяник А. Г.

Сообщение поступило 26.02.88

РЕКЛАМА

Специализированный научно-производственный кооператив «Монитор» при Центре НТМ «Кредо» г. Львова по умеренным договорным ценам ведёт поставку кросс-систем для отладки программного обеспечения контроллеров на базе микропроцессорных серий K580ИК80, KP1816BE48, KP1816BE51 в составе: кросс-ассемблера, редактора-компоновщика, программного отладчика и дизассемблера для инструментальных PDP и IBM совместимых ЭВМ; операционные системы: РАФОС, ФОДОС, RT-11, MS-DOS.

Также предлагается к внедрению и адаптации программного пакета «Паук» поддержки однородной или неоднородной сети контроллеров на базе микропроцессоров вышеуказанных серий и центральной PDP-подобной ЭВМ.

Заказы направлять по адресу: 290044, г. Львов-44, абонентский ящик № 8863

Телефоны: 42-56-83, 34-29-42, 42-86-36

УДК 681.3

В. П. Дьяконов, Т. А. Калаева, С. В. Шилович

БИБЛИОТЕКА АЛГОРИТМОВ И ПРОГРАММ ДЛЯ НАУЧНО-ТЕХНИЧЕСКИХ РАСЧЕТОВ НА ПЕРСОНАЛЬНЫХ ЭВМ

В Смоленском филиале МЭИ создана библиотека прикладных программ и алгоритмов. Ее основой являются алгоритмы и программы, описанные в справочнике*. Более 300 программ*, реализованных на языке Бейсик микроЭВМ «Электроника ДЗ-28», хранятся на магнитофонных кассетах. Это резко ограничивает возможность тиражирования библиотеки программ. В связи с этим в настоящее время библиотека программ полностью реализована на версии БЕЙСИКа ОС РАФОС для вычислительных комплексов ДВК2М и ДВК3. Библиотека (с расширениями) размещается на двух гибких магнитных дисках и может быстро копироваться.

Описание алгоритмов и программ в сочетании с доступностью книги*, изданной массовым тиражом в 300 000 экз., делает данную библиотеку практически завершенным программным продуктом. Библиотека может использоваться для проведения различных массовых научно-технических и инженерных расчетов. Будет работа по дальнейшему расширению библиотеки включением в нее новых пакетов прикладных программ: арифметических операций с произвольной точностью, оптимизации функций многих переменных различными методами, сортировки данных, расчета электронных компонентов и цепей и др.

В ближайшее время планируется перевод библиотеки на версию БЕЙСИКа ПЭВМ IBM PC, что позволит применять программы для совместимых с IBM PC моделей ПЭВМ: ЕС1840, ЕС1841, «Искра 1030», Правец (БНР) и др. Имеются также версии библиотеки для массовых зарубежных ПЭВМ ZX-Spectrum (фирмы Sinclair Research Ltd.), FX-702 (фирмы Casio) и Aquarius (фирмы Mattel Electron). Проводится работа по созданию аналогичной библиотеки программ на языке Паскаль и реализации части программ совместно с элементами графического вывода результатов их работы.

Библиотека программ на языке БЕЙСИК РАФОС (дискетная версия) может передаваться заинтересованным организациям в порядке, предусмотренном действующими правилами о передаче научно-технической информации.

В библиотеках программ*, к сожалению, статистически неизбежны ошибки и неточности. Обнаружены они в программах 3.53 (NEXT I с конца строки 80 нужно перенести в начало строки 90), 3.56 (NEXT M с конца строки 160 нужно перенести на отдельную строку 165), 5.9 ($P=W \times K$ в строке 280 следует заменить на $W=P \times K$), 6.27 (дает неверные результаты для нечетных V), 7.29 (в контрольном примере $R_n=5000$ Ом, $a=0,98$, $R_{вх}=34,946$ Ом и $R_{вых}=253428,6$ Ом) и П5.9 (в строке 120 в выражении $Z \times Z/N$ О знак/ следует заменить на \times). Автор приносит извинения за допущенные неточности и готов оказать письменные консультации пользователям, встретившим затруднения в использовании программ. Обнаруженные неточности устраняются в других версиях библиотеки.

214013, Смоленск, Энергетический пр., 1, СФМЭИ,

каф. Промышленной электроники; тел. 6-77-88 и 9-11-30

Сообщение поступило 15.05.88

* Дьяконов В. П. Справочник по алгоритмам и программам для персональных ЭВМ.— М.: Наука, 1987.— 240 с.

УДК 681.3

В. А. Зайцев, С. Н. Кротов, С. Д. Прошкин,
А. В. Савченко

МУЛЬТИМИКРОПРОЦЕССОРНАЯ СИСТЕМА СПЕКТРАЛЬНОГО АНАЛИЗА СИГНАЛОВ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ

Предлагаемая система спектральной обработки сигнала основана на использовании дополнительных процессорных плат в микроЭВМ с мультипроцессорной архитектурой. Обработка данных идет параллельно на нескольких процессорах, что позволяет спектрально анализировать сигналы одновременно по нескольким (1...8) каналам по алгоритму быстрого преобразования Фурье (БПФ) для 2^n ($n = 1...12$) точек с частотой дискретизации сигнала до 20 кГц, в том числе в реальном масштабе времени с частотой дискретизации до 400 Гц. При поступлении данных с АЦП с частотой более 400 Гц формируется очередь входных данных с их последующей обработкой в порядке поступления.

Рассмотрим основные принципы построения системы обработки аналогового сигнала, организацию программного обеспечения (ПО) дополнительных процессорных плат, протокол межпроцессорного обмена.

Принципы построения системы

Система спектрального анализа сигналов разработана для мультипроцессорной микроЭВМ с разделяемой памятью. Каждая процессорная плата имеет свою локальную память и шину, подключена к общей шине и имеет доступ к общей памяти. Наличие локальной памяти позволяет снизить нагрузку на общую шину и повысить эффективность межпроцессорного обмена. Система спектральной обработки сигналов использует дополнительные (стандартные) процессорные платы быстрого преобразования Фурье (ПБПФ), подключаемые к микроЭВМ.

Каждая ПБПФ обрабатывает один аналоговый канал. В состав ПБПФ входит процессор 18086 [1, 2], процессор с плавающей запятой 18087 [2], локальная память ОЗУ (256 Кбайт) и ПЗУ (64 Кбайт), триггер запуска от внешнего сигнала, контроллер прерываний, порт связи с центральным процессором, 12-разрядный АЦП с временем преобразования 25 мкс, управляемый контроллером прямого доступа к памяти (ПДП) с программируемыми аналоговыми фильтрами на входе, и контроллер общей шины. Одновременно к микроЭВМ можно подключить до восьми ПБПФ.

Для реализации единого механизма обмена данными между процессорными платами используется концепция «логического канала». Логический канал представлен в общей памяти набором управляющих структур данных, служащих для организации и синхронизации межпроцессорного обмена.

0	1	2	4	6	8	10	байт
свободно/ занято	код команды	код возврата	адрес КВ / задания	всего бло- ков ИД	всего бло- ков КФ		

Рис. 1. Контрольный блок ПБПФ:
ИД — исходные данные, КФ — коэффициент Фурье

По фиксированному адресу в общей памяти находится таблица указателей на управляющие структуры данных (контрольные блоки), специфичные для каждой процессорной платы и вида предоставляемых платой услуг. Доступ к таким структурам данных осуществляется через индекс в таблице — «номер логического канала». Структура контрольного блока (КБ) ПБПФ представлена на рис. 1. Каждая ПБПФ создает свой логический канал, номер которого ПБПФ определяет согласно физическому номеру платы.

При включении питания микроЭВМ одновременно с запуском ОС центральным процессором (ЦП) запускается ПО каждой ПБПФ, записанное в ПЗУ. Программа, осуществляющая быстрое преобразование Фурье, выполняется процессором в фоновом режиме. Обслуживание запросов от контроллера ПДП, триггера запуска и ЦП осуществляется по сигналам прерывания, генерируемым этими устройствами.

Сигнал прерывания от ЦП вырабатывается при записи данных в порт связи с ЦП. Программа обработки прерывания от ЦП в зависимости от значения кода, записанного в КБ, выполняет одну из программ ПЗУ ПБПФ, называемую командой платы. Сигнал прерывания от контроллера ПДП вырабатывается по завершении ввода блока данных с АЦП. Прерывание от триггера запуска используется для ввода данных по внешнему событию и синхронизации ввода данных на нескольких платах одновременно.

При решении задачи обработки сигналов с использованием дополнительных процессорных плат существенно то, что наращивание числа обрабатываемых каналов и быстроедействие системы производится подключением дополнительной ПБПФ. Например, расчет функции автокорреляции с помощью одной ПБПФ осуществляется следующим образом: прямое и обратное преобразование Фурье выполняются ПБПФ, результат возводится в квадрат ЦП. Однако, имея две ПБПФ, скорость обработки можно увеличить почти вдвое: прямое преобразование Фурье выполняется первой ПБПФ, результат возводится в квадрат ЦП и, далее, пока вторая ПБПФ производит обратное преобразование Фурье, первая ПБПФ вычисляет прямое преобразование Фурье уже для следующего блока входных данных.

Организация ПО процессорной платы

Единицей работы ПБПФ является «задание», определяющее совокупность ресурсов и режимов, выделяемых процессу. Одновременно на ПБПФ допускается открыть до 16 заданий. ПБПФ способна выполнять специальные команды сбора и обработки данных в рамках указанного задания. Для каждого задания устанавливаются режим ввода и параметры обработки данных (рис. 2). Возможны два источника данных для обработки с помощью ПБПФ: данные для обработки могут быть сформированы каким-либо процессом в общей памяти и выбираются из нее в соответствии с заданным адресом; блок данных (блок ИД) — в локальной памяти ПБПФ аналоговым каналом. Процесс формирования блоков ИД аналоговым каналом в дальнейшем именуется ПДП-процессом.

После формирования блоки ИД снабжаются меткой времени, набором параметров, описывающих тип данных и правила обработки данных блока, и ставятся в очередь на обработку. Строится единая очередь блоков ИД всех заданий данной ПБПФ. Обработка включает в себя сглаживание данных блока ИД с использованием функций Хэннинга, Хэмминга или треугольного окна [3, 4]. Быстрое преобразование Фурье как прямое, так и обратное выпол-

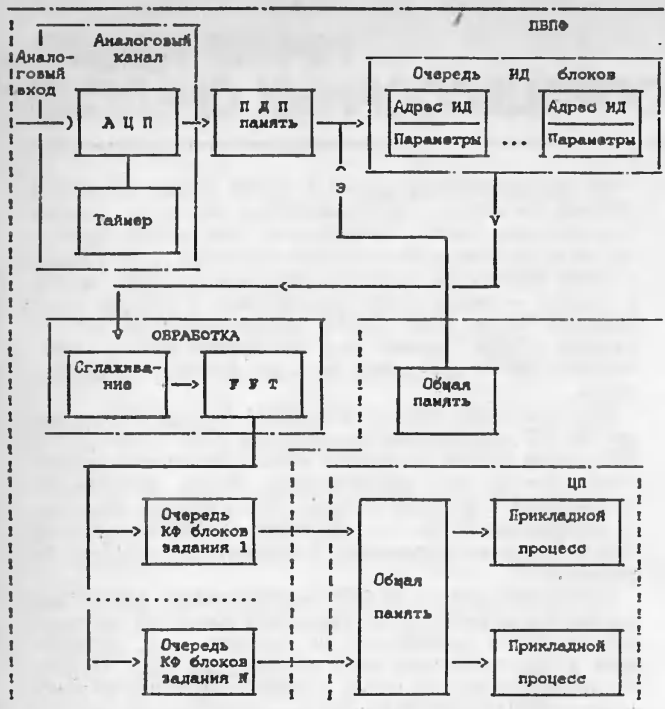


Рис. 2. Последовательность обработки сигнала ПБПФ

няется для 2^n ($n=1 \dots 12$) точек с представлением чисел в формате с плавающей запятой с двойной точностью (64-разряда).

В результате обработки очереди блоков ИД, которая производится фоновой программой, в локальной памяти ПБПФ формируются блоки значений коэффициентов Фурье (блоки КФ). Блоки КФ также снабжаются метками времени, которые копируются из соответствующих блоков ИД. Из блоков КФ формируются очереди для каждого задания отдельно.

Режимы выполнения задания

Задание можно выполнять в одном из следующих четырех режимов:

ONESHOT — однократное формирование ИД блока по запросу ЦП, данные вводятся аналоговым каналом;

AUTORESTART — автоматический переход к формированию и обработке очередного блока ИД после завершения предыдущего цикла, данные вводятся аналоговым каналом;

EXTR — запуск преобразования от внешнего сигнала. Данные вводятся аналоговым каналом. Возможен одновременный запуск преобразования на нескольких ПБПФ путем подачи синхронизирующего импульса, вырабатываемого одной из ПБПФ, на остальные платы;

MULTIBUS — данные выбираются из памяти в соответствии с заданным адресом.

В режимах **ONESHOT**, **AUTORESTART**, **EXTR** можно копировать данные, полученные аналоговым каналом, из блоков ИД в блоки КФ, не производя над ними преобразование Фурье. Одновременно на плате допускается открыть до 16 заданий. При установке режимов необходимо соблюдать следующие правила: режимы **ONESHOT**, **AUTORESTART**, **EXTR** взаимоисключающие, т. е. нельзя на одной ПБПФ одновременно открыть задания в двух различных (из трех, указанных выше) режимах; допускается открыть только одно задание на плате в режиме **AUTO-**

RESTART или **EXTR**. Указанные ограничения вызваны тем, что ПДП-процесс не может выполняться одновременно для нескольких заданий.

Команды ПБПФ

Плата способна выполнять ряд специальных команд (см. Приложение 1). Первой командой любого задания должна быть команда **OpenJob**, открывающая задание. Перед запуском этой команды необходимо создать в общей памяти для данного задания КБЗ и записать его адрес в КБ. Контрольный блок задания создается программой пользователя с описанием режима ввода и обработки данных в рамках задания (рис. 3).

```
typedef struct {
    char sem, /* семафор */
        jobid; /* номер задания */
    int result, /* код возврата */
        qrdB, /* ИД блоков в очереди данного задания */
        qfdb; /* КФ блоков в очереди данного задания */
    union {
        int jt; /*режим работы (используется командой OpenJob)*/
        struct { /* используется командой GetFFT */
            struct { /* смещение */
                unsigned short offset; /* сегмент */
            } deat; /* 32-разрядный адрес буфера */
            int ampno; /* число считываемых значений */
            int strno; /* номер первого считываемого знач. */
        } gt;
        struct { /*метка времени, записывается командой GetStamp*/
            char hour,min,sec,tenth; /* часы,минуты,секунды,
                десятиые доли секунд */
        } st
        struct { /* параметры обработки ИД блока,
            используется командой SetPma */
            char power; /* число точек в ИД блоке */
            float cutfreq; /* частота среза фильтра */
            int wtype; /* тип окна */
            int dir; /*тип преобразования (прямое ПФ,
                обратное, без преобразования) */
        } stdaa;
        struct { /* используется командой Outport */
            int addr; /* адрес порта */
            int val; /* посылаемое значение */
        } portout;
    } opara
} Jobvs;
```

Рис. 3. Контрольный блок задания (в терминах языка Си)

Команда **OpenJob** записывает адрес созданного КБЗ (содержимое 4...7 байт КБ, см. рис. 1) во внутреннюю таблицу. В случае успешного выполнения команда **OpenJob** возвращает в элемент КБ значение номера задания. При дальнейшей работе с тем же заданием в КБ вместо указанного на соответствующий данному заданию КБЗ пишется только его номер. При одновременном открытии на плате нескольких заданий для каждого из них следует создать отдельный КБЗ.

Любая команда ПБПФ запускается через ЦП путем выдачи данных в порт связи с ЦП. Вырабатывается сигнал прерывания, по которому процессор платы ПБПФ начинает выполнять команду. Через КБ передается код команды и номер задания, через КБЗ — параметры для данной команды. В ходе выполнения команды ПБПФ, исключая **GetFFT**, процессор платы оперирует лишь с данными из локальной памяти ПБПФ. Завершается выполнение команды ПБПФ формированием кода возврата, значение которого позволяет контролировать правильность исполнения команд.

Интерфейс с прикладными программами и протокол межпроцессорного обмена

Для удобства работы с ПБПФ создана интерфейсная библиотека **LIVFFT** с подпрограммами на языке Си [5].

(см. в Приложении 2). Пример программы, работающей с ПБПФ и использующей подпрограммы библиотеки, дан в Приложении 3.

Подпрограммы библиотеки делятся на две группы: основные, включающие при выполнении запуск команды ПБПФ (их имя начинается с `fft_`), и вспомогательные, служащие для синхронизации работы с ПБПФ. При открытии задания подпрограммой `fft_openjob` для него инициализируются КБЗ и структура `JOBV` (рис. 4). Во все остальные подпрограммы библиотеки указатель на структуру `JOBV` входит в качестве параметра.

```
typedef struct
{
    HOSTCB * hostcbptr; /* указатель на КВ */
    JOBVC * jobcbptr; /* указатель на КВЗ */
    int cardid; /* номер ПБПФ */
    int result; /* код возврата */
}
```

Рис. 4. Структура `JOBV` (в терминах языка Си)

Последовательность выполнения основных подпрограмм библиотеки `LIBFFT` следующая:

- ожидание, пока семафор `sem` в КБЗ установится в «1» (ожидание завершения выполнения предыдущей FFT-команды данного задания) и установка его в «0»;
- установка параметров задания в КБЗ, если требуется;
- ожидание, пока семафор `sem` в `HOSTCB` установится в «1» (ожидание завершения выполнения текущей FFT-команды любого задания), а затем установка его в «0»;
- запись в `HOSTCB` кода команды и номера задания соответственно;
- запись «0» в порт ПБПФ (собственно запуск FFT-команды).

Сброс флагов `sem` производится программами ПБПФ после выполнения команды платы. Основные подпрограммы библиотеки фактически ожидают выполнения предыдущей подпрограммы и лишь после этого производят требуемое действие.

При использовании вспомогательных подпрограмм библиотеки необходимо учитывать, что к моменту ее исполнения предыдущая команда платы может еще не завершить свое выполнение. Например, выполнение фрагмента программы

```
fft_getnext (job);
i=nfdb (job)
```

может дать значение `i=0` или `i=1`, если в очереди данного задания находился один блок КФ. Выполнив

```
fft_getnext (job);
waitfft (job);
i=nfdb (job),
```

всегда получим значение `i=0`.

Заключение

Система спектрального анализа сигналов реализована на мультипроцессорной микроЭВМ `Labtam 3015`, имеющей в своей конфигурации две процессорные платы на базе микропроцессора `8086` — ЦП и плату — контроллер графического дисплея с высоким разрешением, а также плату на базе микропроцессора `Z80` для организации ввода-вывода на магнитные диски.

Так как программы, выполняющие БПФ, команды ПБПФ, очереди блоков ИД и КФ хранятся в локальной памяти, нагрузка на общую шину незначительна. Это позволяет подключать одновременно несколько ПБПФ без заметного уменьшения скорости обработки данных каждой ПБПФ. В результате удалось осуществить спектральный анализ сигналов одновременно по нескольким (до восьми) каналам по алгоритму быстрого преобразования Фурье для

2^n ($n=1...2$) точек в реальном масштабе времени с частотой дискретизации до 400 Гц.

Система спектрального анализа сигналов разработана в Институте экспериментальной кардиологии ВКНЦ АМН СССР, где успешно применяется для обработки физиологических сигналов. В дальнейшем предполагается модифицировать систему, что позволит в качестве программы обработки сигналов использовать программу пользователя, динамически загружаемую командой ПБПФ в ее локальную память.

Приложение 1

Команды процессорной платы

ПБПФ способна выполнять 13 команд:

`OpenJob` — открыть задание;

`StartDma` — запустить ПДП процесс;

`StopDma` — остановить ПДП процесс;

`FlushDma` — очистить очереди блоков ИД;

`GetFFT` — прочитать блок КФ, стоящий первым в очереди;

`GetStamp` — прочитать метку времени блока КФ, стоящего первым в очереди;

`GetNext` — продвинуть очередь блоков КФ на один блок;

`CloseJob` — закрыть задание (освободить ресурсы, отведенные под задание);

`SetDma` — установить параметры ПДП процесса и обработки данных (частота среза фильтра, размер блока данных для выполнения БПФ, прямое или обратное БПФ, тип сглаживающей функции);

`Mbus` — начать обработку данных в режиме `MULTIBUS`;

`Vers` — считать номер версии программного обеспечения;

`Ground` — считать с АЦП значение, соответствующее его заземленному входу;

`Outport` — послать байт в локальный порт ПБПФ (используется в программах калибровки и тестирования платы).

Приложение 2

Подпрограммы (библиотека `LIBFFT.LIB`)

`fft_closejob` — закрыть задание, освободив выделенные ресурсы;

`fft_flushdma` — очистить очередь;

`fft_getfft` — считать данные после преобразования Фурье;

`fft_getnext` — перейти к следующему блоку КФ;

`fft_getstamp` — считать метку времени;

`fft_ground` — калибровка АЦП;

`fft_mbus` — запуск преобразования Фурье. Данные из массива в памяти;

`fft_openjob` — открыть задание;

`fft_outport` — послать байт в порт;

`fft_setdma` — установить параметры преобразования для данного задания. Данные с АЦП;

`fft_startdma` — запуск ПДП-процесса;

`fft_stopdma` — останов ПДП-процесса;

`fft_vers` — считать номер версии программного обеспечения;

`nfdb` — определение числа блоков КФ;

`nrdb` — определение числа блоков ИД;

`retr` — определение кода возврата;

`waitbuf` — ожидание, пока не будет сформирован хотя бы один блок КФ;

`waitfft` — ожидание завершения основной команды.

Приложение 3

Пример программы, работающей с платой ПБПФ и использующей подпрограммы библиотеки `LIBFFT`.

```

typedef struct {
    unsigned short offset,
                    segment;
} ADDR32;

typedef struct {
    char    power,
           wtype,
           dir;
} PARAM;

#define N        256 /* число точек в блоке для БПФ */
#define CARD_NO 0   /* номер ПБПФ */
#define ONESHOT 1   /* режим */
#define NO_CARD -1  /* отсутствует ПБПФ с указанным номером */
double xr[N];      /* буфер для результата БПФ */
ADDR32 fdata;      /* адрес для результата БПФ */
PARAM * hdata;     /* указатель на параметры обработки */
JOBID * job;

main()
{
    int c, i;
    JOBID * fft_openJob();

    initdbh(); /* подготовить параметры обработки */
    job = fft_openJob(CARD_NO, ONESHOT); /* открыть задание */
    i = (int)job;
    if( i==0 || i==NO_CARD )
    {
        printf("Open Failure\n");
        if( i == NO_CARD ) printf("Cannot find FFT card");
        exit();
    }
    i = retr (job); /* код возврата команды OpenJob */
    if( i != 0 ) { printf ("Open Failure\n"); exit(); }

    fft_setdma ( job, hdata ); /*установить пар. обработки*/
    fft_startdma ( job ); /*стартовать ввод одного блока*/
    for ( ;; )
    {
        waitfft ( job ); /*ожидание завершения команды GetNext */
        waitbuf ( job ); /* ожидание формирования блока КФ */
        fft_startdma ( job ); /* стартовать ввод одного блока */
        fft_getfft ( job, &fdata, 0, N/2); /* записать по адресу
        fdata N/2 коэффициентов Фурье, начиная с 0 */
        waitfft (job); /*сидание завершения команды GetFFT */
        outpbuf (); /* печать коэффициентов Фурье */
        printf ("\nenter RETURN to continue, any key to exit");
        c = getch ();
        if ( c != '\012' )
            { fft_closeJob ( job ); exit (); }
        fft_getnext (job); /* перейти к следующему блоку КФ */
    }
}

#define FREQ 2000
#define HAMMING 2
#define DIRECT 0
#define CUT_FREQ 4 /* соответствует частоте среза 500Гц */

initdbh() /* подготовить параметры обработки */
{

```

```

hdata = (PARAM *) malloc( sizeof(PARAM) );
hdata->dir = DIRECT; /*тип преобразования */
hdata->cutfreq = CUT_FREQ; /* частота среза фильтра */
hdata->power = N; /* число точек в ИД блоке */
hdata->wtype = HAMMING; /* тип окна */
hdata->pacer = FREQ; /* частота дискретизации */

fdata.offset = (int) xr; /* адрес (сместение) для коэф.Фурье*/
fdata.segment = getds(); /* адрес (сегмент) для коэф.Фурье */

outpbuf () /* печать коэффициентов Фурье */
{
    int i;
    for ( i=0; i<N/2; i++ )
        printf ("xr[%d]: re_part=%1f ia_part=%1f\n", i, xr[i*2], xr[i*2+1]);
}

```

Данная программа производит прямое преобразование Фурье аналогового сигнала и выводит на экран терминала значение КФ. Данные вводятся в режиме ONESHOT.

Приложение 4

Производительность ПБПФ приведена в табл. 1 и 2.

Таблица 1

Время выполнения команд ПБПФ

Команда	Время выполнения, ис	Примечание
OpenJob, StartDma	Не более 3 3	200 мс при изменении номера фильтра
StopDma, FlushDma	1,5 3	
GetFFT	1...70	В зависимости от числа точек, пересылаемых из блока КФ
GetStamp, GetNext	Не более 1 1	
CloseJob	4	
SetDma	1,5	
Mbus	2	
Ground	200	
Outport	1	
Vers	1	

Таблица 2

Скорость выполнения преобразования Фурье

Тип преобразования	Число точек						
	64	128	256	512	1024	2048	4096
Время выполнения							
Прямое преобразование Фурье без предварительного сглаживания, с	0,1	0,2	0,5	1,1	2,4	5,1	11,5
с предварительным сглаживанием, с	0,1	0,2	0,5	1,2	2,6	5,5	12,0
Обратное преобразование Фурье, с	0,1	0,2	0,5	1,0	2,3	4,9	11,0
Пересылка блока данных без выполнения преобразования Фурье, с	0,01	0,02	0,02	0,03	0,05	0,09	0,16

Максимальная частота дискретизации

Прямое преобразование Фурье, Гц	—	520	500	450	420	400	360
---------------------------------	---	-----	-----	-----	-----	-----	-----

Максимальная частота дискретизации допускает выполнение прямого преобразования Фурье (без предварительного сглаживания) в реальном масштабе времени в режиме AUTORESTART. При работе с частотой, превышающей указанную, данные будут поступать быстрее, чем обрабатываться, что приведет к росту очереди блоков ИД и необходимости периодически очищать ее с помощью команды FlushDma.

Телефон 414-66-56, Москва

ЛИТЕРАТУРА

1. Liu Yu-Cheng, Gibson G. A. Microcomputer systems; the 8086/8088 family; architecture, programming, and design.— Prentice — Hall: Englewood Cliffs, 1984.
2. Gorsline G. W. 16-bit modern microcomputers. The Intel 18086 family.— Prentice — Hall: New Jersey, 1985.
3. Rao K. R. Fast transforms: algorithms, analyses, applications.— New York: Academic Press, 1982.
4. Бендат Дж., Пирсол А. Измерение и анализ случайных процессов. Пер. с англ.— М.: Мир, 1974.
5. Kernighan B. W., Ritchie D. M. The C programming language.— Prentice — Hall: Englewood Cliffs, 1978.

Статья поступила 18.02.88

УДК 681.327

А. В. Бабаянц, Г. Е. Фрезе, А. С. Вартанов, Г. Н. Волянский, М. М. Гадильшин

МИКРОПРОЦЕССОРНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ ПЕРИОДИЧЕСКИМИ ПРОЦЕССАМИ БИОТЕХНОЛОГИЧЕСКИХ ПРОИЗВОДСТВ

Устройство «Биоцикл МП-01» разработано на базе элементов КТС ЛИУС-2 [1] как типовое изделие для контроля и цифрового управления периодическими технологическими процессами с учетом опыта промышленного применения серии ранних разработок устройств комплекса «Биоцикл» [2]. Оно предназначено для АСУТП различных отраслей промышленности (см. таблицу).

Типизация устройства основана на полном использо-

Технические характеристики устройства

Параметр	Значение	Наличие аппаратного контроля или формирования сигналов
Дискретные сигналы 12 В:		
от технологических датчиков	40	Да
контроля положения ПИМ	96	Да
контроля положения регулирующих исполнительных механизмов (РИМ)	16	Нет
дополнительные	24	Нет
Непрерывные сигналы 0...5 мА	44	Да
от технологических датчиков		
Дискретные сигналы 24 В (или пневматические 0,14МПа) для управления ПИМ	96	—
Непрерывные сигналы 0...5 мА для управления РИМ	16	—
Объем памяти (МПС ЧМС/МПС КРУ), Кбайт		
оперативной энергозависимой	8/8	
оперативной	16/64	
перепрограммируемой	32/64	

Примечание. Параметры контроля и управления распределены поровну между двумя МПС КРУ

вании возможностей конструктивных элементов для организации связей с технологическим объектом (ТО) с характерным для периодических процессов численным преобладанием каналов дискретного управления над каналами аналогового управления.

Для решения вопросов горячего резервирования, самодиагностики, максимизации функций контроля и управления, уменьшения стоимости, с одной стороны, и конструктивных ограничений, с другой, выбрана структура устройства, включающая в себя микропроцессорную станцию (МПС) человеко-машинной связи (ЧМС); встроенный дисплей с клавиатурой; две МПС контроля регулирования и управления (КРУ) технологическим объектом; панели аппаратных средств резервирования (АСР).

Каждая МПС КРУ предназначена для контроля и управления одним ТО (при небольшой информационной мощности ТО допускается управление несколькими объектами, но с общим режимом — ручным или автоматическим для позиционных исполнительных механизмов (ПИМ) ввода и обработки сигналов от дискретных и аналоговых датчиков технологических параметров, программно-логического управления (ПЛУ) ПИМ и непосредственного цифрового управления (НЦУ) — регулирования.

МПС ЧМС накапливает технологическую и системную информацию от МПС КРУ устройства, отображает ее на дисплее и распоряжается работой МПС КРУ. Управление работой МПС ЧМС осуществляется вводом директив с функциональной клавиатуры.

Панель АСР предназначена для ручного резервирования функций контроля, регулирования и управления МПС КРУ. Функция НЦУ резервируется специально разработанными электронными ПИ-регуляторами.

В устройстве предусмотрены диагностика работы, ручной и автоматический перезапуск МПС после их останова, взаимное горячее резервирование функций, ручное или автоматическое включение аппаратных резервных регуляторов, диагностика работы аналого-цифровых преобразователей (АЦП); возможность подключения к аппаратным регуляторам независимых контуров измерения параметров.

Резервирование функции контроля и регулирования (от 0 до 100 %) основано на использовании остатка соответствующих каналов после проектной привязки ТО к устройству (за счет программно-аппаратной координации управления ПИМ достигнуто 100 %-ное резервирование данной функции МПС). Контроль работоспособности, перезапуск и резервирование функций МПС осуществляются через разработанные элементы резервирования.

В качестве программного обеспечения устройства можно использовать комплекс СЦУ [3], а также программные модули, реализующие функции централизованного контроля, ЧМС, ПЛУ и НЦУ на ассемблере, настраиваемые под конкретный объект управления.

Устройство может быть использовано для управления ТО с непрерывной технологией.

364060, Грозный, пр. Кирова, 2, НПО «Промавтоматика»; тел. 7-03-04

ЛИТЕРАТУРА

1. ГСП. Отраслевой каталог, т. 4, вып. 4. «Комплекс технических средств для локальных информационно-управляющих систем на базе микросхем с повышенной степенью интеграции и микропроцессоров (КТС ЛИУС-2). Общая часть». — М.: ЦНТТГЭИ приборостроения, 1981, № 10.
2. Опришко А. А., Бабаянц А. В., Ханукаев Я. А. и др. Микропроцессорный комплекс «Биоцикл» для управления периодическими процессами ферментации // Приборы и системы управления. — 1983. — № 4. — С. 29—30.
3. Коневцов В. А., Казаченко А. П., Бабаянц А. В. МикроДАТ Программные средства цифрового управления. — М.: ЦНИИГЭИ приборостроения, 1985, вып. 5, 6. — 70 с.

Статья поступила 29.12.87

МОДУЛЬ СИСТЕМЫ ОБРАБОТКИ ДАННЫХ РЕАЛЬНОГО ВРЕМЕНИ

Развитие методов и средств цифровой обработки информации позволяет создавать цифровые приборы и системы для решения задач обработки данных в реальном масштабе времени в медицинской и технической диагностике, сейсмологии, акустике, обработке изображений, которые целесообразно выполнять на основе унифицированных узлов. Быстродействия современных ЭВМ недостаточно для решения перечисленных задач. Для повышения возможностей применяют периферийные векторные (сигнальные) процессоры [1, 2], работающие под управлением ЭВМ и имеющие собственные быстродействующие ЗУ данных, скоростные АЛУ и набор микропрограмм для выполнения элементарных операций над массивами (векторами) данных. Кроме элементарных векторных операций процессоры могут иметь микропрограммы часто встречающихся алгоритмов обработки, например свертку, быстрое преобразование Фурье, формирование и преобразование графических изображений. Программирование можно осуществлять на языках высокого уровня.

Для выполнения обработки в реальном масштабе времени недостаточно быстродействия векторного процессора, поэтому задачи распределяются между несколькими процессорами, объединенными в систему. Обмен данными целесообразно осуществлять с помощью скоростных средств непосредственно между векторными процессорами. Ниже описан модуль, выполненный на базе быстродействующего периферийного процессора (БПП) «Электроника МТ-70» и ЭВМ «Электроника МС 1201.02» [3], установленной в корпусе БПП.

Процессор МТ-70 не имеет внешних каналов ввода-вывода, поэтому для организации внешнего обмена данными доработано БПП.

МикроЭВМ содержит интерфейсы, достаточные для подключения внешних устройств при тестировании модуля, отладке управляющих и загрузке отла-

женных программ. При подключении к встроенной микроЭВМ дисплея и накопителя на гибких магнитных дисках модуль можно использовать как отдельное программируемое быстродействующее устройство обработки информации, управляемое ОС ФОДОС, РАФОС, ОС ДВК или RT-11. Отлаженные управляющие программы могут работать в микроЭВМ без ОС и внешних устройств. Они загружаются управляющей системой через интерфейсы модуля с помощью программ аппаратных загрузчиков системного ПЗУ.

ЭВМ МС 1201.02 устанавливается вместо коннекторов связи непосредственно в корзине БПП; на посадочные места коннекторов подведены сигналы К ПИТ, К ПОСТ, К ОСТ и К ПРТ, вырабатываемые внутренними источниками питания БПП; контакты сигналов предоставления прерывания прямого доступа замкнуты. В контроллере БПП доработана плата и ПЗУ микропрограмм; установлены дополнительно: два разъема (вход и выход) типа СНО 53-60-23, приемный и передающий регистры, выходные передатчики, дешифратор поля установок микрослова контроллера и две пары триггеров управления прерываниями ввода и вывода. При этом сохранены все функции контроллера, максимально использованы имеющиеся узлы и система микрокоманд.

Ввод-вывод осуществляется независимо, асинхронно, параллельно, последовательно, с квитированием. Асинхронность достигается с помощью системы прерываний процессора контроллера [1] (используются запросы ЗП7 — ввод и ЗП3 — вывод). Запрос ЗП7 предусмотрен в БПП для регенерации памяти управляющей ЭВМ, ЗП3 резервирован.

Регистры ввода (РГ ВХ) и вывода (РГ ВЫХ) подключены в место соединения выходов буферного регистра (БД) контроллера с М-входами микросхем процессора контроля (см. рисунок). При внешнем вводе данных выходы регистра БД (К589ИР12) переключаются в Z-состояние, а регистра РГ ВХ — открываются.

Для адресации буферов ввода и вывода в памяти данных БПП используются резервные регистры процессора контроллера БПП RG3 (канальный адрес 174006) и RG9 (канальный адрес 174022) соответственно. Так как их всего два, для адресации буфера и счета слов используется один и тот же регистр. Одна из границ буфера привязывается к границе адресного пространства памяти данных БПП. Размер буфера регулируется управляющей программой по значению нефиксированной границы. При вводе-выводе слова со-

держимое регистра увеличивается на два. Переполнение регистра адреса процессора обозначает достижение границы буфера. Использование в микропрограммах вывода и ввода прямого и инверсного адресов позволяет создать два буфера, один из которых привязан к началу адресного пространства (ввод), второй — к концу (вывод). Прерывания внешнего ввода и вывода разрешаются при записи управляющей программой адреса в регистры адреса ввода и вывода процессора контроллера; сброс происходит при переполнении этих регистров.

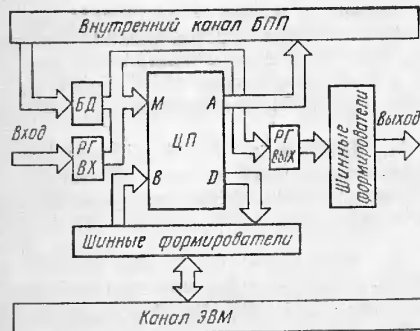
Сигналы разрешения прерывания на разьемах внешнего ввода и вывода объединяются монтажным ИЛИ с соответствующими сигналами источника или приемника так, чтобы обмен происходил, когда оба устройства готовы к нему. Для начального запуска обмена при разрешении прерывания вывода устанавливается и требование его прерывания. После заполнения выходного регистра во время вывода на выходной разъем поступает импульсный сигнал «Ответ», который на входном разьеме принимающего устройства используется как сигнал «Запрос», требующий прервать ввод. При вводе слова на входном разьеме приемным устройством вырабатывается сигнал «Ответ», который в свою очередь поступает на вход «Запрос» выходного разьема передающего устройства и требует прервать вывод. Подача запросного сигнала на вход модуля обеспечивает инициализацию прерываний со стороны источника не имеющего средств для этого способа квитирования.

Регистры ввода и вывода, выходные передатчики данных и сигналы квитирования выполнены на микросхемах серии К531. Входные цепи и цепи квитирования согласованы на 120 Ом наборами резисторных делителей. Входные регистры могут использоваться в режиме постоянного приема или защелкиваться.

Запросы ЗП3 и ЗП7 переключают ПЗУ микропрограмм на вторую (дополнительную) страницу, где записаны микропрограммы внешнего ввода-вывода.

Микропрограммы контроллера приведены в табл. 1. В табл. 2 дана расшифровка обозначений действий дешифратора поля установок, микропрограмм внешнего ввода-вывода.

Тестирование обмена ввода-вывода осуществляется на одном модуле при соединении входного и выходного разъемов кабелем связи из комплекта БПП. Суммарная скорость обмена по вводу и выводу более 400 Кслов/с. Максимальная скорость ввода определяется длиной микропрограммы ввода и составляет 500 Кслов/с. Однако при работе с такой скоростью во время ввода не должно быть обращений к контроллеру БПП со стороны ЭВМ, так как использование для ввода запроса ЗП7 с наивысшим приоритетом подавляет обслуживание контроллером БПП операций ввода и вывода со-



Структурная схема модуля

«Микропроцессорные средства и системы» № 5, 1989

N	ПСА	Поле ЦПЭ	С	К	М	ПУС	Поле	
							установок	ветвлений
Вывод								
3	0	R9→RA	1	1	0	БУС	УСТ ТР РЕЖИМА	ЗАПР КАН
0	1	НОП	1	1	0	ТР ЗАГР БД	НОП	НОП
1	2	—»—	1	1	0	БУС	ВЫВОД	—»—
2	4,5	R9+2→R9	0	1	0	БУС, К=1	НОП	ВЫХ ПЕРЕНОС ЦП
4	12	НОП	1	1	0	БУС	СБР РАЗР ВЫВ	НОП
5	37	—»—	1	1	0	—»—	СБР ТР СТР2	—»—
Ввод								
7	10, 11	R3+2→R3	0	1	0	БУС, К=1	СБР ТР РЕЖИМА	ВЫХ ПЕРЕНОС ЦП
10	12	НОП	1	1	0	БУС	СБР РАЗР ВВОДА	НОП
11	14	R3→AC	1	1	0	—»—	НОП	—»—
14	15	NEG(AC)	1	1	0	—»—	—»—	—»—
15	16	AC→RA	1	1	0	—»—	—»—	—»—
16	17	M→AC	1	1	0	—»—	ВВОД	ЗАПР КАН
17	20	NEG(AC)	1	1	0	Т ЗАПР КАН	НОП	НОП
20	12	AC→RA	1	1	0	БУС	—»—	—»—
12	37	НОП	1	1	0	—»—	СБР ТР СТР2	—»—
37	37	—»—	0	0	0	—»—	—»—	—»—

Примечания: ПСА — поле следующего адреса; С, К, М — поля управления входным переносом, маской и адресацией регистров процессора контроллера соответственно; ПУС — поле условной синхронизации; номера микрокоманд в графах N, М, К и ПСА восьмеричные; R3, R9 — регистры ЦПЭ; RA и AC — выходной регистр А и аккумулятор ЦПЭ; NEG — инверсия; НОП — нет операции; БУС — безусловная синхронизация; Т ЗАГР БД — синхронизация по загрузке регистра БД, Т ЗАПР КАН — по запросу внутреннего канала БПП контроллером; К=1 — расширение поля входного переноса, обозначающее добавление 1 в операции сложения; ЗАПРОС КАНАЛА в поле ветвлений — расширение поля установок (по ветвлениям — НОП).

стороны канала ЭВМ, что может привести к незапланированным прерываниям работы ЭВМ.

Начало ввода-вывода N отсчетов иницируется записью числа NA—2X X (32 767—N) в регистр адреса ввода (вывода) контроллера БПП. Конец ввода-вывода определяется при равенстве нулю содержимого этого регистра. Ниже приведены примеры фрагментов программ на ассемблере и Фортране для управления вводом и выводом.

```
MOV NA, α#174006
I α: TST α#174006
CALL IPOKE (α#174022, NA)
IT-IPEEK (α#174022)
IF (IT.NE.0) GO TO 1
```

равный 177 550, и передать со стороны управляющей ЭВМ команды В и LA для вызова программы абсолютного загрузчика. Далее нужно передать управляющую программу с носителем ЭВМ управления системой в модуль обработки по команде монитора. COPY/BIN ИМЯ. LDA DEV., где DEV — имя устройства и драйвера связи ЭВМ управления системой с модулями обработки.

ИНИЦИАЦИЯ НАЧАЛА ВВОДА
ОЖИДАНИЕ ОКОНЧАНИЯ ВВОДА
ИНИЦИАЦИЯ НАЧАЛА ВЫВОДА
ОЖИДАНИЕ ОКОНЧАНИЯ ВЫВОДА

Программы можно также загружать через последовательный интерфейс (ИРПС) микроЭВМ модуля обработки. Для этого кроме действий, описанных выше, необходимо после команд В и LA передать команды: 157 776/(ПС) 177 560 для изменения в микроЭВМ модуля адреса устройства, с которого будет загружаться управляющая программа.

Для загрузки управляющих программ в модуле используется один из трех интерфейсов встроеной микроЭВМ, поэтому свободные интерфейсы можно применять для органи-

Таблица 2
Действия дополнительного дешифратора поля установок

Функции	Код	Действия
НОП	0	Нет операции
ВЫВОД	1	Переписка данных с выхода регистра БД в РГ ВЫХ
ВВОД	2	Занесение данных с выхода РГ ВХ в ЦП при закрытом выходе регистра БД
СБР РАЗР ВЫВ	3	Сброс триггера разрешения прерывания вывода
УСТ ТР РЕЖИМА	4	Установка триггера режима в положение, соответствующее циклу вывода в канале БПП
СБР ТР РЕЖИМА	5	Установка триггера режима в положение, соответствующее циклу ввода в канале БПП
СБР РАЗР ВВОДА	6	Сброс триггера разрешения прерывания ввода
СБР ТР СТР2	7	Сброс триггера управления страницами ПЗУ микропрограмм контроллера БПП

Отлаженные программы компонируются в формате абсолютной двоичной загрузки (ИМЯ. LDA) с включением специального модуля SIMRT, необходимого для создания программ, работающих без ОС по команде монитора. LINK/LDA/INCLUDE ИМЯ, при выполнении которой в ответ на запрос компоновщика LIBRARY SEARCH? необходимо ввести имя модуля α SIMRT.

Программы в модули системы загружаются с помощью интерфейсов микроЭВМ модуля и устройств связи ЭВМ управления системой. Для загрузки через байтовый радиально-параллельный интерфейс (ИРПР) необходимо переключателями установить адрес ИРПР на плате микроЭВМ,

зации связей (поперечных или типа «обратная связь») между модулями системы или для создания дополнительных каналов подключения устройств индикации и диагностики. 252056, Киев, Политехническая, 16, КБ «Штурм»; тел. 441-11-77

ЛИТЕРАТУРА

1. Процессор быстродействующий периферийный «Электроника МТ-

70». 15-ВУМС 8-028. Техническое описание ПГЦМ 3. 770.001 ТО.— 1980.

2. Березовский А. В., Козлачков И. А., Коршевер И. И., Павлов С. А. Сигнальный процессор СП-8 // Автометрия.— 1986.— № 4.

3. МикроЭВМ «Электроника НМС 11100.1» (МС 1201). Техническое описание 0.305.019 ТО1. Статья поступила 30.09.87

УДК 681.327

М. Г. Мифтахов, В. С. Бортасевич, Г. Р. Мифтахова

МНОГОПРОЦЕССОРНАЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩАЯ СИСТЕМА С КАНАЛОМ ПДП ДЛЯ ГЕОФИЗИЧЕСКИХ ИССЛЕДОВАНИЙ СКВАЖИН

Информационно-управляющая система для геофизических* исследований скважин (ГИС), разработанная во ВНИИ Геофизических исследований совместно с ВНИИ Нефтепромгеофизики, представляет собой бортовую программно-управляемую лабораторию, входящую в состав самоходной каротажной станции** — СКС-2УУ, размещенной на одном шасси автомобиля повышенной проходимости. СКС-2УУ включает в себя:

информационно-управляющую систему (ИУС) с мультипроцессорной ОС реального времени (МОС РВ) ГИС; спуско-подъемное оборудование с набором технологических датчиков; систему энергоснабжения.

Станция работает совместно с агрегированным комплексом измерительных преобразователей — скважинным (АКИПС). Информационный обмен между АКИПС и ИУС осуществляется с помощью программно-управляемой цифровой телеметрической системы, имеющей характеристики, приведенные ниже:

Способ обмена	Синхронный двусторонний по принципу команда-ответ
Код данных	Манчестер-II
Скорость передачи, Бод, не более	80

* Геофизические методы разведки — исследование земной коры физическими методами с целью поиска и разведки полезных ископаемых.
** Каротаж — геофизические исследования скважин, выполняемые с целью изучения геологических разрезов и выявления полезных ископаемых.

Рис. 2. Структурная схема одноплатной микроЭВМ.

Для передачи информации и питания скважинного прибора используется одна жила кабеля и броня. Функции каротажной станции нового типа:

- управление процессом спуско-подъема и сбора геофизической информации, поступающей от АКИПС;
- регистрация всей геофизической и технологической информации;
- визуализация информации для оперативного контроля качества результатов измерения;
- первичная обработка поступающей информации для упорядочения данных по глубинам и масштабам;
- проверка работоспособности измерительных каналов в процессе и после проведения каротажа.

ИУС — распределенная многопроцессорная система с магистральной структурой (рис. 1). МикроЭВМ ИУС объединяются через системную магистраль И-41. Наиболее «уязвимое место» данной структуры — наложение по времени запросов от различных процессоров на доступ к системной магистрали.

Максимальная пропускная способность магистрали достигается благодаря использованию высокоскоростного канала передач данных с применением ПДП. Для ГИС выделены три подсистемы: сбора геофизических данных и управления процессом каротажа, документирования и визуализации и хранения геофизических данных.

Основа рассматриваемых подсистем — одноплатные микроЭВМ на базе микропроцессора КР580ИК80А (рис. 2).

Основные характеристики микроЭВМ:

Тактовая частота, МГц	2
Внутреннее ОЗУ, Кбайт	16
ПЗУ, Кбайт	4
Внутренние УВВ	КР580ВИ53, КР580ИК57, КР580ВН59
Число внешних УВВ, не более	128

Все одноплатные микроЭВМ ИУС взаимозаменяемы и отличаются индивидуальным адресом, «защитным» в Д5 (рис. 3), и программы обслуживания соответствующего периферийного оборудования. Конструктивно микроЭВМ выполнены на двусторонней печатной плате размерами 233×220 мм (двойная плата Е2) с тремя разъемами типа СМП59-96.

В любой операции обмена данными на системной магистрали всегда участвуют два устройства, связанных между собой как микроЭВМ-Инициатор и микроЭВМ-Абонент. МикроЭВМ на системной магистрали представляют собой порты ввода-вывода.

МОС РВ ГИС, размещенная на

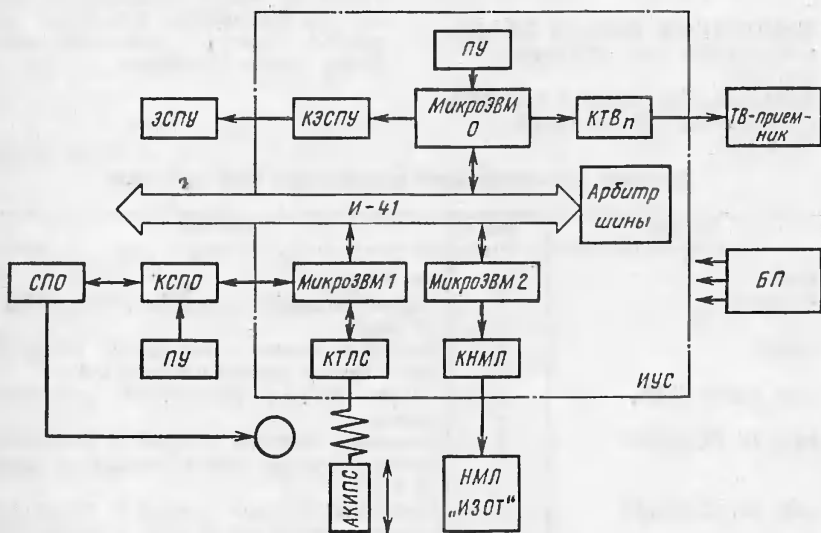
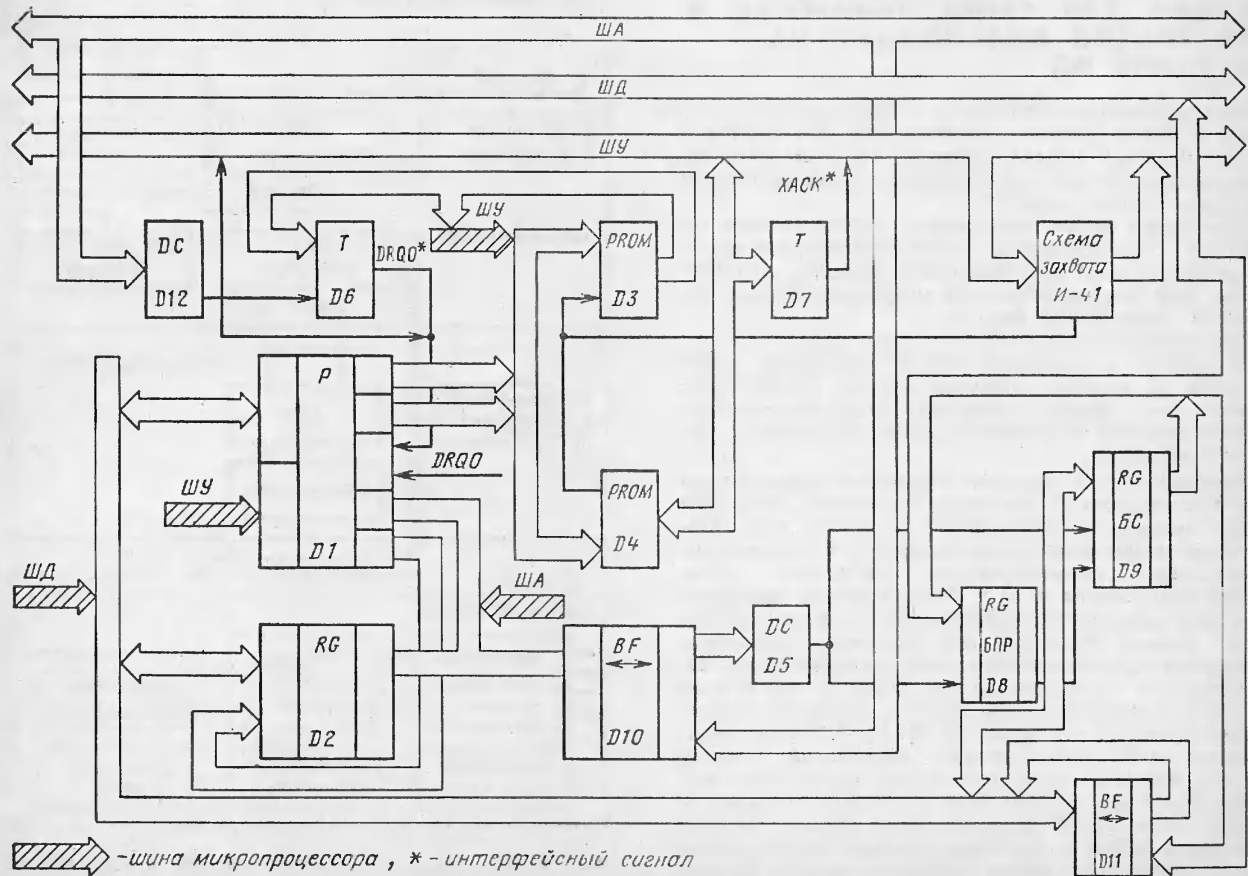
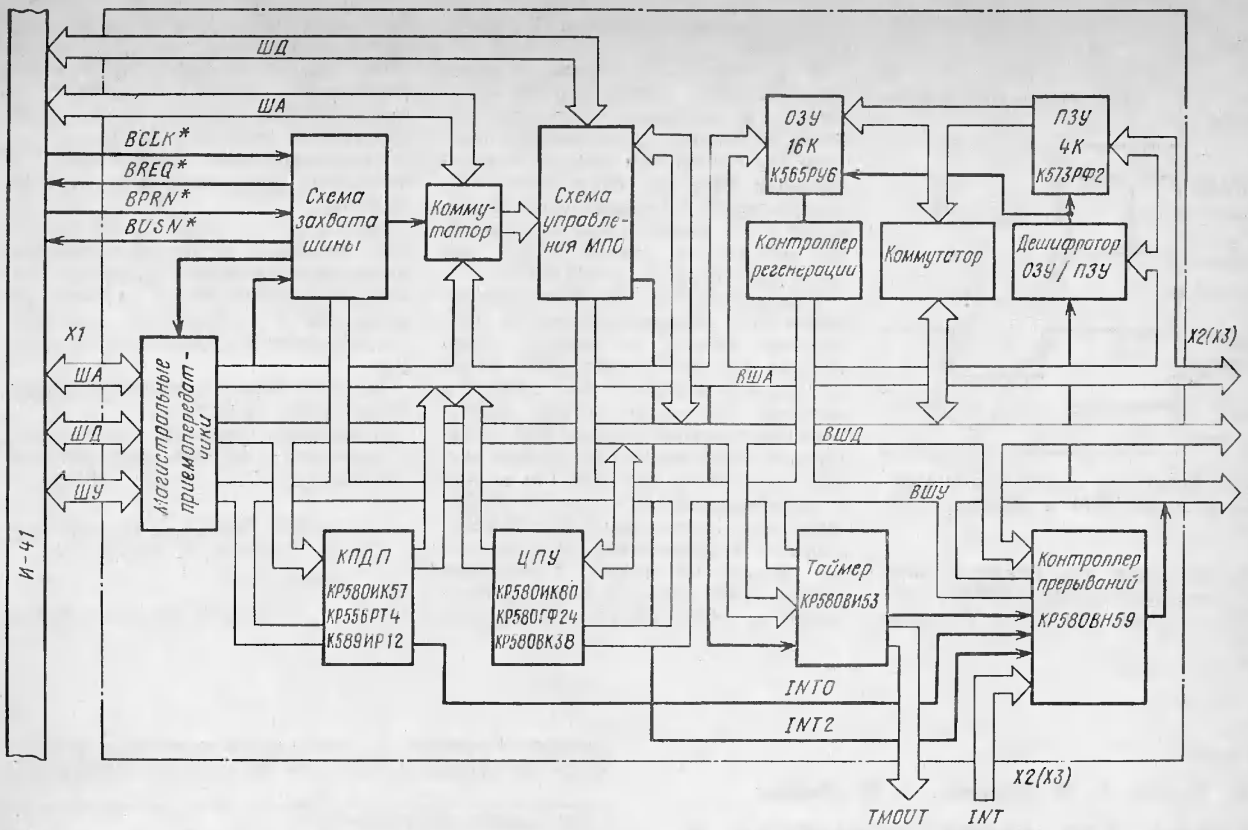


Рис. 1. Информационно-управляющая система в составе станции:

КТВ — контроллер ТВ-приемника, КЭСПУ — контроллер электростатического печатающего устройства, КТЛС — контроллер телеизмерительной линии связи, КНМЛ — контроллер накопителя на МЛ, КСПО — контроллер спуско-подъемного оборудования, АКИПС — агрегированный комплекс измерительных преобразователей скважинный, ПУ — пульт управления, БП — блок питания

Рис. 3. Контроллер межпроцессорного обмена:

→ — шина микропроцессора, • — интерфейсный сигнал →



▨ - шина микропроцессора, * - интерфейсный сигнал

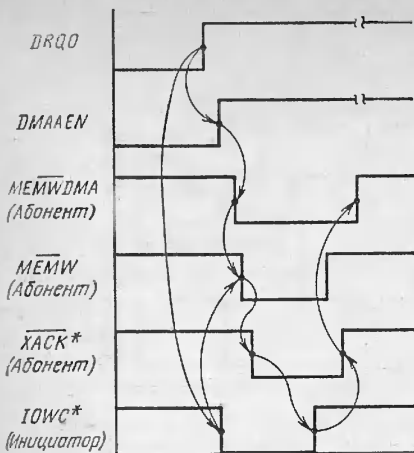


Рис. 4. Временная диаграмма синхронизации микро ЭВМ в режиме ПДП

ПЗУ, реализует два основных типа межпроцессорного обмена (МПО) микроЭВМ: запуск задачи в Абоненте

BEGIN) и обмен массивами (PUTOZY, GETOZY).

В режиме ПДП обмен данными осуществляется асинхронно по отношению к сигналам синхронизации шины по сигналу подтверждения передачи XACK Абонента (рис. 4). В контроллерах МПО по концу счета (КС) контроллера ПДП инициируется прерывание INT0; микроЭВМ готовы к обмену данными в режиме ПДП или происходит переход на BEGIN.

В подпрограмме обработки прерывания INT2 Абонент выделяет из БПР команду обмена и личный номер Инициатора, настраивает БИС ПДП на прием и переводит флаг DRQ0 в активное состояние (старт ПДП).

Первоначальный вариант ИУС представлял собой многопроцессорную систему межрегистровой передачи данных с максимальной скоростью 10 кБод/с (при скорости передачи ТЛС=25 кБод, скорость геофизических исследований на уровне 100 м/час). Применение канала ПДП повышает скорость передачи до 400 кБод/с, разгружает

системную магистраль и увеличивает производительность системы. Кроме этого, рассматриваемая программно-управляемая станция в отличие от существующих серийно выпускаемых каротажных станций позволяет:

- автоматизировать диагностику и подготовку аппаратуры к проведению каротажа;
- проводить за одну спуско-подъемную операцию многопараметрические исследования скважин, используя применение кодоимпульсной цифровой телесистемы и комплексных приборов;
- тиражировать каротажные диаграммы;
- выводить символьную и графическую информацию на дисплей;
- программно управлять подъемником;
- проводить экспресс-интерпретацию результатов ГИС после каротажа.

452620, БАССР, г. Октябрьский, ул. Горького, 1, ВНИИГИС, тел. 5-79-58

Статья поступила 4.08.88

УДК 681.326

А. П. Иванов, А. Н. Осьмаков, В. В. Лысенко

СИСТЕМА ДЛЯ СБОРА, ОБРАБОТКИ И РЕГИСТРАЦИИ ИНФОРМАЦИИ НА КАССЕТНОЙ МЛ

При исследовании естественного электромагнитного поля Земли в целях прогноза землетрясений для работы в жестких условиях полевых экспериментов необходимо накопление больших объемов информации и их длительная непрерывная регистрация.

В настоящей работе представлена многоканальная система для сбора, обработки потока информации и хранения результатов на долговременном носителе, построенная на базе широко известного микропроцессорного набора БИС серии КР580 (рис. 1).

Блок преобразования кодирует входной аналоговый сигнал в формате с плавающей запятой. Под каждое число отводится 12 двоичных разрядов (восемь соответствуют мантиссе числа, четыре — порядку). Такой формат обеспечивает широкий динамический диапазон устройства при малой разрядности.

Информация регистрируется с помощью трехдвигательного лентопротяжного механизма кассетного типа. При записи последовательности двоичных чисел в виде двухчастотной манипуляции сигнал кодируется в регистраторе: единица одним колебанием частоты в 5 кГц, нуль — двумя колебаниями частоты 10 кГц. Информация на магнитную ленту записывается последовательным кодом одновременно по четырем дорожкам. Для увеличения амплитуды сигнала при воспроизведении частотно-модулированной информации последовательности режим записи каждой из частот оптимизируется по отдаче, т. е. каждой частоте соответствует свой ток записи [1, 2]. На МЛ также записывается стробирующий сигнал, содержащий меандр 1,25 кГц, период которого соответствует восьми битам или одному байту информации, причем его срез совпадает с первым битом (число периодов стробирующего сигнала равно числу байтов информации). Кодовая последовательность и стробирующий сигнал разделяются при считывании с помощью простых аналоговых фильтров, не вносящих

фазовые искажения. Наиболее распространенные дефекты магнитного носителя, такие как «выпадения» и паразитная

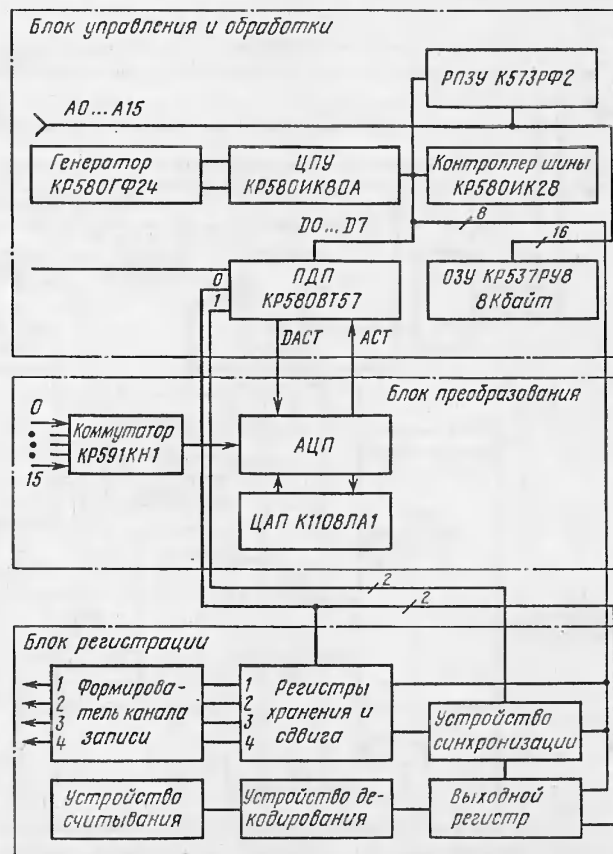


Рис. 1. Блок-схема системы для сбора и регистрации информации

амплитудная модуляция, практически не влияют на запись низких частот порядка 1,25 кГц при скорости 4,76 мм/с. Поэтому потеря нескольких битов (период частоты 5 или 10 кГц) не вызывает сбоя синхронизации.

С помощью оптимизации режима записи по отдаче усилитель на линейном участке АЧХ (0,5...20 кГц) воспроизводит частоты 1,25; 5; 10 кГц с одинаковой амплитудой, которая при считывании 5 и 10 кГц позволяет построить дешифратор кодовой последовательности, не чувствительный к паразитной амплитудной модуляции в пределах $\pm 50\%$ и детонации $\pm 20\%$. При такой низкой чувствительности к помехам плотность записи порядка 120 бит/мм получается даже на лентопротяжном механизме IV класса. Практическая работа с накопителем показала, что при считывании после записи на новую предварительно размагниченную кассету, потери информации нет. Исключение составляют кассеты с механическими дефектами.

Передача данных в блок регистрации организуется пачками по четыре байта, которые пересылаются в регистры интерфейса, соответствующие четырем каналам записи. В режиме записи и считывания магнитофон имеет наивысший приоритет.

Данный способ кодирования, применение многоканальной магнитной головки и системы дискретного формирования тока записи повышают плотность информации на магнитной ленте; объем хранимой информации на стандартной компакт-кассете С-90 увеличивается до 9 Мбайт неформатной емкости при непрерывной работе устройства. В условиях полевых экспериментов важно уменьшить энергопотребление, поэтому часть устройства собрана на КМОП-элементах серии К561.

Работой руководит блок системы управления обработкой, который представляет собой микроЭВМ с ограниченной архитектурой. Машина собрана по классической схеме с 8-разрядной шиной данных и 16-разрядной адресной магистралью, которые совместно с линиями управления составляют системную магистраль комплекса. Функции согласования и обслуживания блоков АЦП и магнитофона выполняет контроллер ПДП. Информация из блока преобразования в оперативную память передается по инициативе АЦП по сигналу готовности данных, который поступает на вход требования АСТ; ответный сигнал ДАСТ открывает шину данных АЦП.

Технические параметры системы:

Тактовая частота процессора, МГц	2
Объем, Кбайт	8
ОЗУ,	4
РПЗУ,	9
Объем информации на кассете, Мбайт	4,75
Скорость движения МЛ, см/с	126
Плотность записи информации на МЛ, бит/мм	10 ⁻⁸
Надежность записи, бит	96
Динамический диапазон, дБ	96
Число разрядов АЦП, бит	12
Потребляемая мощность, Вт:	
при заполнении буфера	6
при работе магнитофона	1,5
Время записи буфера, с	0,5
Число входных каналов, не более	16
Последовательная одновременная запись	По 4 каналам

Улучшение параметров конкретной задачи возможно программными средствами. На рис. 2 представлена блок-схема программы, преобразующей поступающие двухбайтные коды оцифровки входного сигнала в более компактный вид и позволяющей без потери точности входной информации увеличить время непрерывной записи на одну кассету в 2—5 раз. Программа иллюстрирует перераспределение функций между аппаратными и программными средствами системы.

Работа программы. Оперативная память системы разбита на четыре области; две временно хранят данные, третья — информацию, подготовленную для вывода на магнитную ленту, оставшаяся часть используется для работы программы. Временные входные буферы поочередно заполняются данными, информация из которых поступает на обработку. По каналу $y(i)$, где $i=1,16$, выбираются

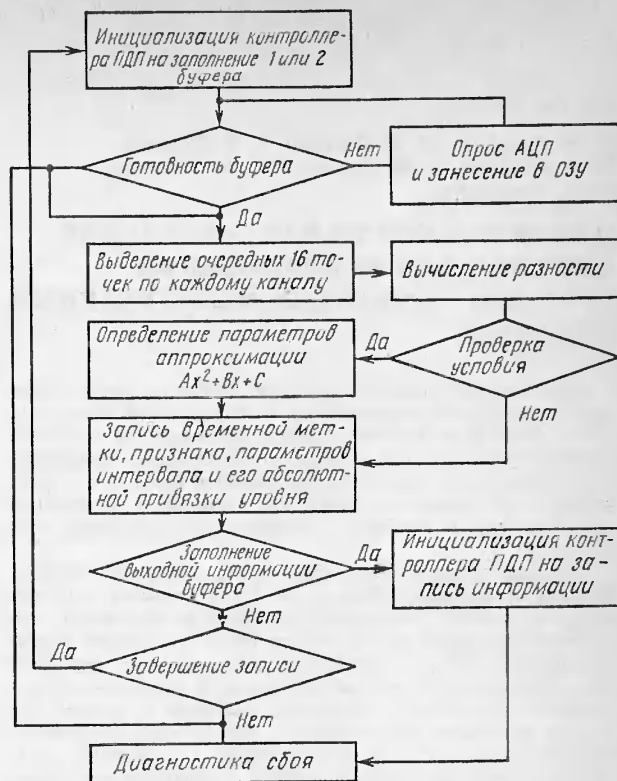


Рис. 2. Алгоритм программы управления комплексом в режиме длительного накопления информации

16 последовательных отсчетов и оценивается разность каждой соседней пары. Ряд считается гладким и может быть аппроксимирован степенным полиномом, если ни одна разность не превышает некоторого числа ϵ . Практика показывает, что достаточно квадратичного полинома, чтобы вносимая ошибка не превышала 1% (число фиксируемых параметров резко сокращается). При быстрых изменениях входных сигналов целесообразно записывать вычисленные разности $\Delta_i = y(i+1) - y(i)$, где $i=1,15$ с начальной привязкой абсолютного значения интервала. Так как для восстановления формы сигнала шаг квантования по времени должен быть много меньше периода сигнала, поэтому порядки величин двух соседних отсчетов одинаковы и разность двух 16-разрядных чисел удастся записать в один байт, т. е. для полного восстановления входного сигнала необходимо зафиксировать вдвое меньшее число байтов информации.

123810, Москва, Б. Грузинская, 10, ИФЗ АН СССР; тел. 254-91-50

ЛИТЕРАТУРА

1. Магнитная запись электрических сигналов: По материалам иностр. период. печати. Пер. А. И. Вичеса.— М.: Энергия, 1967.
2. Богородский Ю. Л. Разрешающая способность систем магнитной записи.— М.: Энергия, 1980.

Статья поступила 1.12.87

УДК 681.3:616.1.072

В. И. Лукьянов, О. Е. Логинов, В. Б. Даньков,
Е. А. Тетерина, С. М. Бородкин, С. М. Блинков,
М. Г. Никандров

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ ИССЛЕДОВАНИЯ ВРЕМЕНИ ДВИГАТЕЛЬНЫХ РЕАКЦИЙ ЧЕЛОВЕКА

Исследование времени реакции (ВР) на зрительные, слуховые и другие раздражения — эффективный метод изучения специфики высшей нервной деятельности человека: определения индивидуальных особенностей, колебаний функционального состояния центральной нервной системы, воздействия различных факторов внешней и внутренней среды, а также различных повреждений головного мозга [1—3].

Система, реализующая этот метод, позволяет изучать простые и сложные реакции по 1... 5 каналам одновременно в режиме автоматизированного эксперимента, что обеспечивает возможность исследования отношений между различными чувствительными и двигательными центрами при одновременной их работе. Адресуя различные раздражения (стимулы) — слуховые, световые и прочие — к левому и правому полушариям и анализируя статистические закономерности времени реакции в различных сочетаниях, можно исследовать важные особенности пространственно-временной организации функционирования мозга.

Информационно-измерительная система (ИИС) для определения ВР (рис. 1) эксплуатируется в лаборатории экспериментальной неврологии Московского института нейрохирургии в течение года. Методика исследования состоит в следующем. Пациент сидит в кресле с подлокотниками, на которых, как и на подставке для ног, укреплены контактные приспособления для регистрации движений рук и ног. Для регистрации движений нижней челюсти на кронштейне подается специальное контактное устройство. При необходимости одновременно регистрируется внешнее дыхание, ЭКГ и т. д. Экспериментатор осуществ-

ляет визуальный контроль из другой комнаты и, если потребуется, связывается с пациентом по телефону. В течение одного сеанса проводится несколько серий по 32... 64 опыта в каждой. Спыт начинается подачей предупреждающего сигнала, после которого, с некоторой временной задержкой (случайной или регулируемой), подается световой или звуковой сигнал-стимул. В ответ пациент должен максимально быстро отреагировать заданным движением рук, ног, челюсти в соответствии с конкретной задачей обследования. Времена реакций по каждому каналу измеряются соответствующими входными измерителями (электромиорефлексометрами ЭМР-01) и вводятся в компьютер для обработки и анализа.

Информация с четырех счетных тетрад каждого рефлексометра подается на входной разъем блока разовых команд (БЛРК) информационно-измерительного комплекса (ИИК) ГАММА. Комплекс представляет собой сочетание функционально и конструктивно законченных модулей, объединенных магистралью передачи команд и данных, применяется в системах оценки функционального состояния пациента.

Принцип работы БЛРК состоит в преобразовании 80 сигналов разовых команд в восемь 10-разрядных двоичных кодов, разделенных во времени, что обеспечивает возможность одновременного подключения до пяти электромиорефлексометров.

Контроллером в ИИК ГАММА служит блок управления и синхронизации, формирующий кадр системы. Блок вырабатывает кадровые синхросигналы KSS, обозначающие начало кадров; тактовые синхросигналы KTSS, предназначенные для опросов блоков ИИК; тактовые синхросигналы TSS — для сопровождения 10-разрядной информации в кадре. Частота следования TSS может составлять 16 384, 8192, 2048, 1024, 512 Гц.

В состав ПЭВМ «Искра 226.6» входит интерфейсный блок БИФ «Искра-015-82», обеспечивающий связь ПЭВМ с внешними устройствами в соответствии с рангом ИРПР согласно ОСТ 25778-77 (интерфейс для радиального подключения устройств с параллельной передачей информации).

Для согласования магистрали ИИК ГАММА с магистралью ИРПР разработано устройство сопряжения (рис. 2). Информация в виде 10-разрядного параллельного кода подается на входы буферного регистра D3...D5, запись в который производится с помощью сигналов TSS блока БУС, задержанных на одновибраторе, собранном на элементах D11...D13, R11, C1. Ввод информации в ПЭВМ синхронизируется с помощью триггера D6 и схемы совпадения D2.1. В начальное состояние триггер устанавливается управляющим сигналом ИФС, вырабатываемым ПЭВМ. С приходом KSS из блока БУС триггер открывается и выдает сигнал, поступающий на вход 1 схемы совпадения. На вход 2 поступают импульсные сигналы с частотой следования F1, равной удвоенной частоте TSS. Стробы источника STR-P, свидетельствующие о достоверности информации, выставленной на линии данных, формируются из отселектированных импульсов частотой f1 одновибратором D2.3, D2.4, D12.4, R13, C3. Удвоение частоты STR-P по сравнению с частотой следования информации необходимо для 2-байтного ввода в ПЭВМ 10-разрядного слова за один тактовый промежуток. При этом в первый байт входят восемь младших разрядов слова, а во второй — два старших. Старшие и младшие разряды коммутируются схемой выборки D11. Элементы D7...D10, D13, D14 служат для инвертирования и согласования сигналов в ПЭВМ. Временная диаграмма работы устройства согласования представлена на рис. 3.

Монотонный характер экспериментов, рассчитанных на сравнительно длинные серии однообразных актов, позволяет передать компьютеру функции подачи предупрежда-

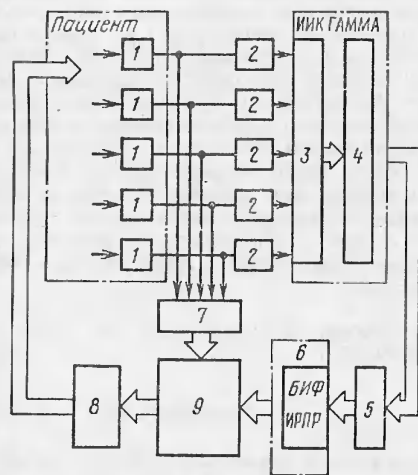


Рис. 1. Структура информационно-измерительной системы: 1 — первичные преобразователи для измерения времени реакции; 2 — электромиорефлексометры ЭМР-01; 3 — блок разовых команд; 4 — блок управления и синхронизации; 5 — устройство сопряжения; 6 — ПЭВМ «Искра 226.6»; 7 — блок отображения состояния первичных преобразователей; 8 — пульт управления; 9 — врач-исследователь

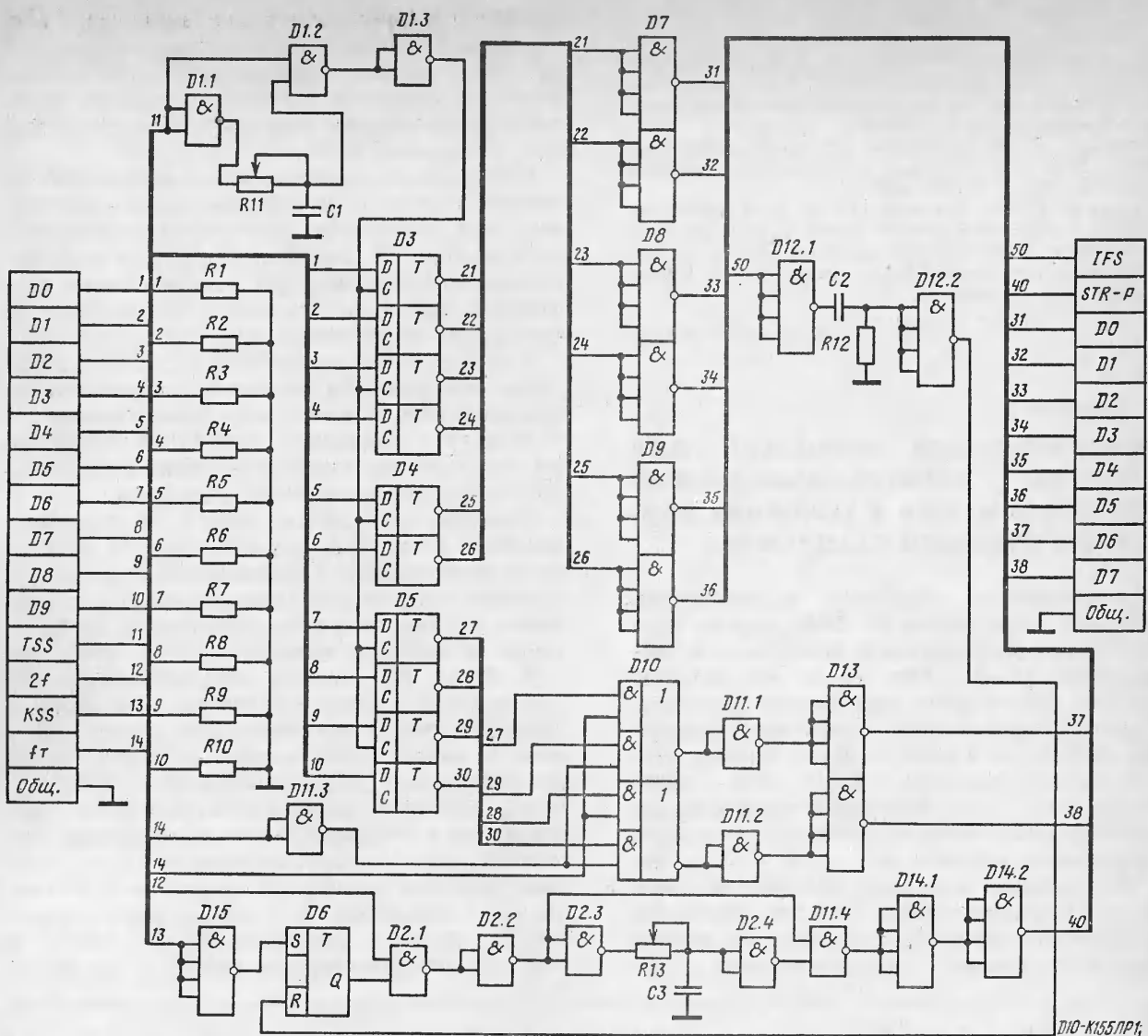


Рис. 2. Электрическая схема устройства сопряжения

ющих команд, оповещения испытуемого об измеренных в каждом акте временных задержках и т. п. Применение ПЭВМ в условиях длительных испытаний дает возможность исследовать параллельно нескольких пациентов (в условиях клиники).

Значения времен реакций с точностью до 1 мс накапливаются в соответствующих массивах; обработка результатов и расчет обобщенных показателей осуществляются по окончании каждой серии измерений. Рассчитываются гистограммы распределения, основные статистические характеристики (средние, дисперсии, коэффициенты асимметрии и другие), а также парные и частные коэффициенты корреляции каналов. Обеспечивается также ввод и сопровождение необходимой паспортной информации и форматный вывод результатов, сохранение их на дискетах для организации архивов.

Годовая эксплуатация ИИС показала ее ценность и перспективность в проведении обследований больных нейрохирургической клиники с целью уточнения диагноза, а также при проведении профотбора летчиков, водителей транспорта, спортсменов.

Телефон 258-81-56, Москва

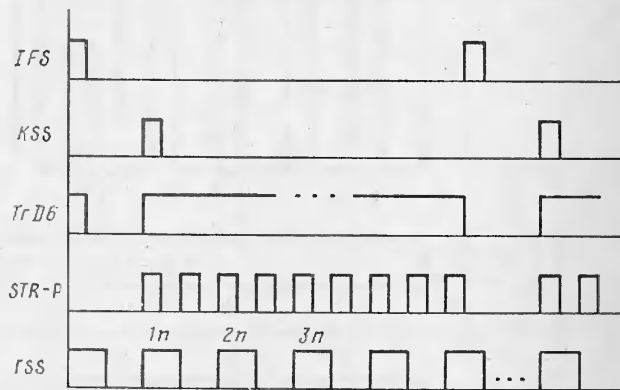


Рис. 3. Временная диаграмма работы устройства согласования

1. Артарян А. А., Блинков С. М. Локализация функций мозга в пространстве и времени. В кн.: Механизмы динамической локализации и компенсации функций мозга.— Ереван, 1986.— С. 199—201.
2. Блинков С. М. К методике раскрытия механизмов мозга. В кн.: А. Р. Лурия и современная психология.— М., 1981.— С. 224—229.
3. Блинков С. М., Москатова А. К. К механизму простой двигательной реакции. Время реакции при попеременном и одновременном движении обеими руками. Тр. Моск. Госуд. Университета. 1986.— Вып. 1. Психологические исследования.

Статья поступила 13.04.88

УДК 681.32

Л. Г. Вайнштейн

ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС ДЛЯ ОБРАБОТКИ ЭЛЕКТРОФИЗИОЛОГИЧЕСКОЙ ИНФОРМАЦИИ В УСЛОВИЯХ НЕЙРОХИРУРГИЧЕСКОГО СТАЦИОНАРА

Математическая обработка электрофизиологической информации на ЭВМ широко применяется при диагностике в неврологии и нейрохирургии [1—3]. Уже много лет ведущие фирмы по производству электронной аппаратуры для электрофизиологических исследований («Nihon Kohder» и «San — Ei» — Япония, «Disa» — Дания, «Nicolett» — США, «Bio — medica» — Италия и др.) выпускают компьютерные системы для получения вызванных потенциалов, спектрального анализа и т. д. В СССР также выпускались подобные накопители (Ф36 и Ф37). Стандартизация методов обработки электрофизиологической информации и выпуск специализированных вычислительных ус-

тройств сопровождаются поиском новых, а также совершенствованием известных методик [4, 5]. Все больше применяется нейромониторинг [6]: интра- и послеоперационный контроль за состоянием нейрохирургических больных с помощью ЭВМ.

Использовать универсальные микроЭВМ (с соответствующими периферийными устройствами) для обработки электрофизиологической информации [7] дешево, удобнее (в ряде исследовательских работ [8] это единственно возможный выход по сравнению со специализированными вычислителями).

Универсальные микроЭВМ с соответствующими программами позволяют обрабатывать электрокардиограммы, электромиограммы и т. д. и этим уменьшить количество аппаратуры, что особенно важно в условиях операционного и послеоперационного отделений.

Отметим, что система [8] и ей подобные несмотря на их большие возможности практически неприменимы в большинстве нейрохирургических стационаров ввиду значительных размеров, стоимости, а также сложности приобретения и наладки комплектующих приборов.

В статье [9] описаны две вычислительные системы для работы в условиях стационара и сформулированы требования к подобным системам. С нашей точки зрения, подобные системы обязательно должны вести архив исходной и обработанной электрофизиологической информации в цифровой форме на магнитных носителях; обрабатывать сигналы не только биоэлектрической активности мозга, но и вегетативных процессов — электрокардиограмм (ЭКГ), фото- и реоплетизмограмм (ФПГ и РЭГ), электромиограмм (ЭМГ) и др; иметь

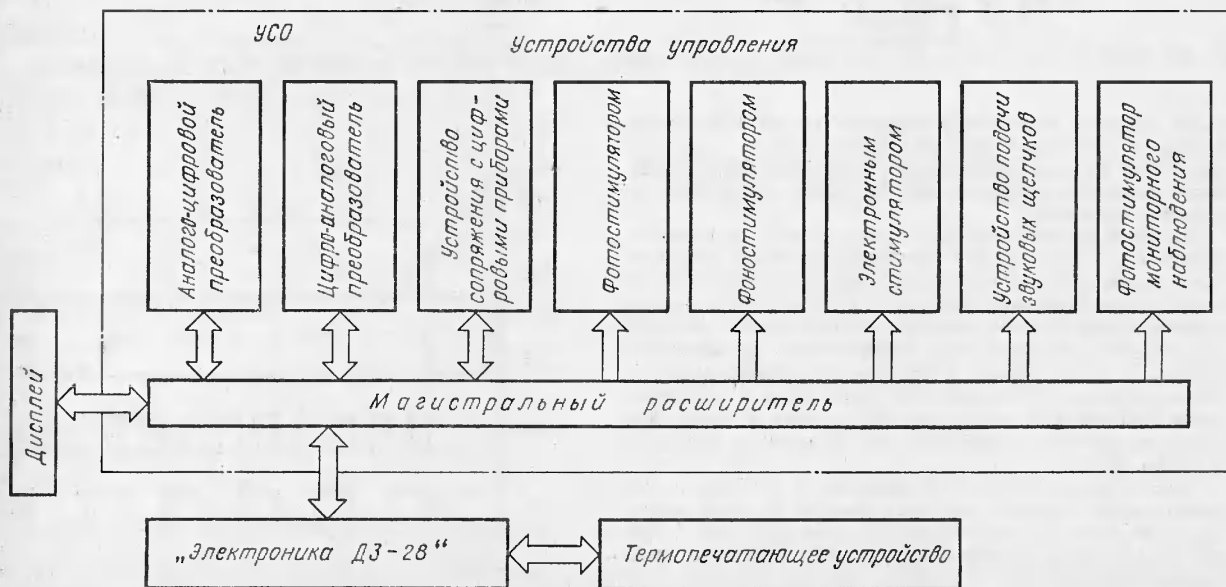


Рис. 1. Подвижный вычислительный комплекс для обработки электрофизиологической информации

многоканальный (не менее 8 каналов [5, 10]) ввод электрофизиологической информации и коммутацию сигналов на входе; управлять электрофизиологической аппаратурой.

Архив в цифровой форме необходим для ответственности данных при развитии методов обработки электрофизиологической информации, статистической обработки, визуального сопоставления в различных масштабах и т. д. Многоканальность [10] и обработка вегетативных процессов нужны для выделения из суммарной биоэлектрической активности мозга артефактов внесозгового происхождения (ЭКГ, ЭМГ, сосудистых пульсаций) и из-за сложности геометрии и структурной организации мозга.

Так как обработка электрофизиологических сигналов требует, как правило, их предварительного усиления и определенной частоты характеристики усилителя (в зависимости от сигнала), целесообразно для усиления использовать полиграфы или электроэнцефалографы с разъемом для подключения вычислительного комплекса.

Необходимость управления электрофизиологической аппаратурой также обусловлена физиологическими требованиями — использованием функциональных нагрузок (звуковой, световой, электрической стимуляции) для оценки функционального состояния мозга.

Вид управления (активный или в режиме прерывания, т. е. пассивный) стимулирующей аппаратурой принципиально важен. Режим прерывания [9] проще реализовать, однако он имеет ряд недостатков: худшую защищенность от артефактов; невозможность установления обратной связи (по отношению к мозгу) и создания особых видов стимуляции (пакетов, стимулов, переменных интервалов и т. п.), необходимых для научных исследований. Если вычислительная система активно управляет стимулирующей аппаратурой, указанные недостатки можно устранить.

В Ленинградском нейрохирургическом институте уже несколько лет эксплуатируются вычислительные комплексы на базе вычислительной системы 15ИПГ32—003 с микроЭВМ «Электроника ДЗ-28».

В состав вычислительного комплекса (рис. 1) входят система подготовки программ (дисплей 15 ИЗ-03—001, микроЭВМ «Электроника ДЗ-28», термopечатающее устройство), серийно выпускаемое устройство связи с объектом — УСО И5М3.852.031 или АЦСКС-1024—001 (магистральный расширитель, одноканальный цифро-аналоговый преобразователь, устройство сопряжения с цифровым вольтметром и частотомером) и специально разработанные устройства: быстродействующий 8- и 16-канальный аналого-цифровые преобразователи

(АЦП), устройства управления фото-, фоно- и электронным стимуляторами, устройством подачи звуковых щелчков, а также фотостимулятор для мониторингового интраоперационного контроля.

АЦП (с максимальной частотой опроса 28 кГц) собран на БИС К1113ПВ1А (рис. 2). Данные преобразования пересылаются в однобайтовом формате (БИС — 10-разрядная, цена младшего разряда — 10 мВ). Суженный диапазон допустимых сигналов на входе АЦП соответствует диапазону амплитуды выходного сигнала большинства типов усилителей электроэнцефалографов, а время на пересылку результата преобразования (почти полностью соответствующее времени преобразования) сокращается вдвое и во столько же раз экономится объем памяти. Максимальная частота АЦП (по одному каналу) — 28 кГц. Номер опрашиваемого канала задается при обращении к АЦП. Частота опроса регулируется программно. Быстродействие АЦП позволяет регистрировать медленные [9] и быстрые электрофизиологические процессы — стволовые вызванные потенциалы [10] и т. п.

Устройства управления стимуляторами представляют собой дешифратор адреса (рис. 3) и преобразователи сигнала ТТЛ-уровня в сигнал требуемой формы и уровня, необходимых для запуска стимулятора.

Фотостимулятор для мониторингового контроля [11, 12] предназначен для подачи с помощью светоизлучательной системы на светодиодах — в виде надеваемых на глаза очков, регулируемых по яркости и длительности вспышек. Также, как устройства управления, он имеет дешифратор адреса, запускающий ждущий мультипликатор с регулируемой длительностью выходного сигнала. Яркость вспышек регулируется ключом, изменяющим ток в цепи питания светодиодов. Устройство подачи звуковых щелчков подобно фотостимулятору для мониторингового контроля.

Описанный вычислительный комплекс использовался совместно с электроэнцефалографом ME-175, EEG-7209 фирмы «Nihon Kohden» — (Япония), ЭЭГ-16—04 (СССР), EEG-16 «Medicor» (ВНР).

Структуру вычислительного комплекса можно наращивать и усложнять — подключать (в составе УСО есть плата сопряжения) НМД, а также организовывать мультипроцессорную вычислительную систему.

Программное обеспечение комплекса включает в себя программы: анализа биоэлектрической активности мозга, анализа вегетативных процессов (сердечного ритма и оценки сосудистого тонуса), обработки результатов спектрального, корреляционного и статистического анализов данных.

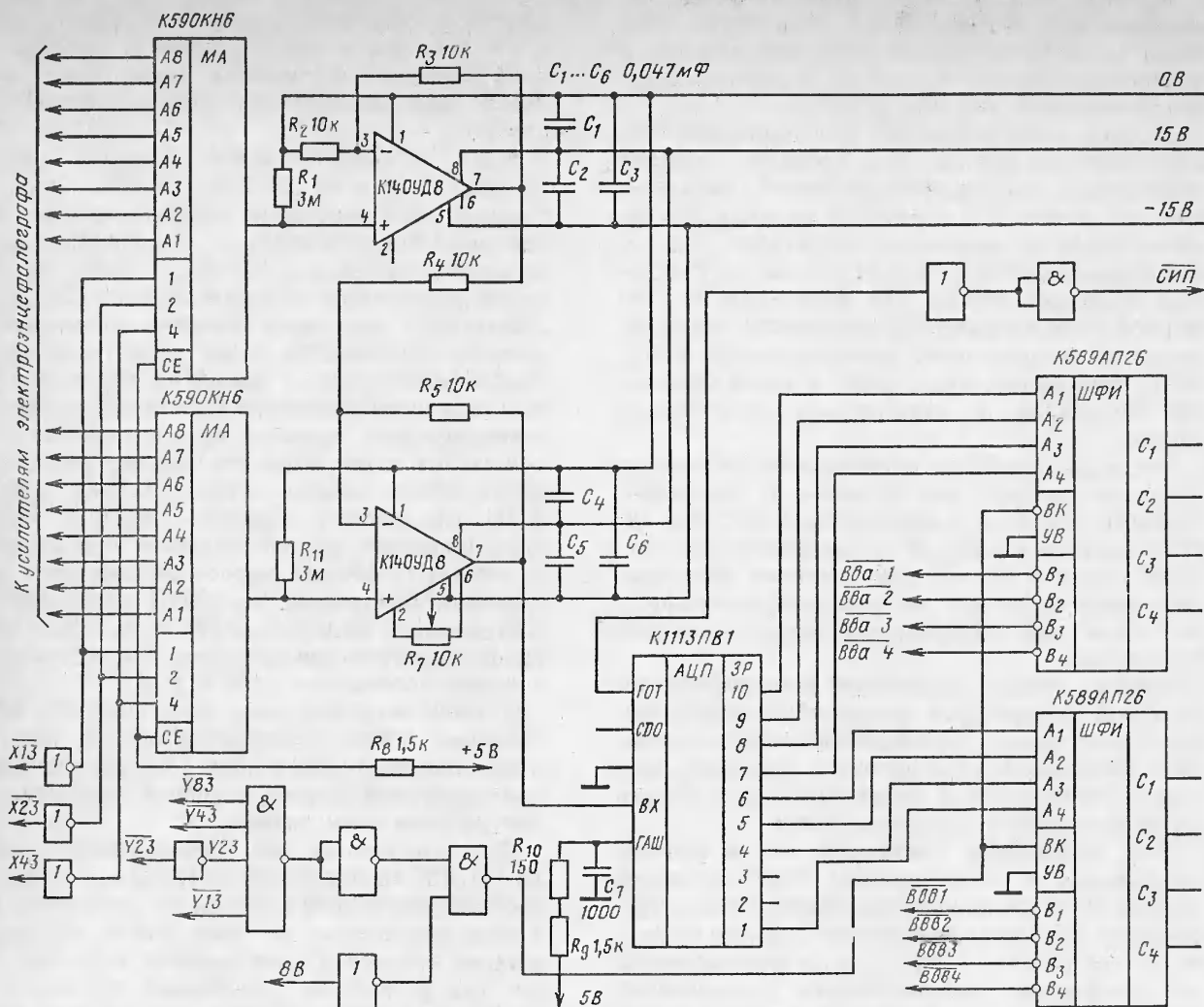


Рис. 2. Схема аналого-цифрового преобразователя

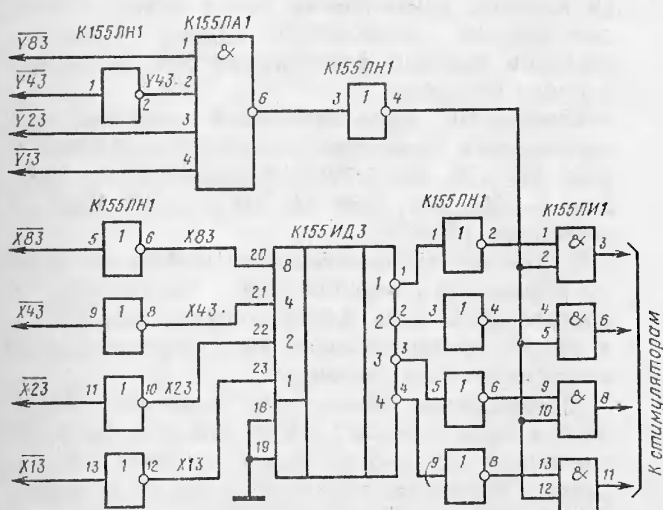


Рис. 3. Схема дешифратора адреса для устройств управления стимуляцией

К первой группе программ относятся программы ввода биоэлектрической активности — спонтанной и вызванной на стимулы (коротко- и длиннотенных) различной модальности (звуковой, зрительной, соматосенсорной). Последние можно регистрировать одновременно во многих отведениях (до 12). Это необходимо при топографическом анализе вызванных потенциалов [5, 13] для оценки функционального состояния различных отделов полушарий мозга.

Программы спектрального анализа с использованием быстрого преобразования Фурье и корреляционного анализа можно использовать для любых временных рядов (представление данных в одно- и двубайтовом формате).

Программы анализа вегетативных процессов — сердечного ритма, сосудистого тонуса — служат и для оценки функционального состояния мозга, так как эти процессы имеют цент-

ральную регуляцию. В программе анализа сердечного ритма помимо статистических характеристик (среднего значения, дисперсии, асимметрии, эксцесса, максимального и минимального значений) строится гистограмма зависимости длительности последующего сердечного цикла от предыдущего.

Все программы написаны на ассемблере, оптимизированы по быстродействию и объему памяти, выводят исходную и обработанную информацию на магнитную ленту, компактные кассеты, т. е. можно вести архив.

Результаты обработки выводятся на дисплей, термопечатающее устройство, самописец (или электроэнцефалограф, если он имеет вход по постоянному току), осциллограф.

Телефон 279-51-37, Ленинград

ЛИТЕРАТУРА

1. Биопотенциалы мозга человека. Математический анализ / Под ред. В. С. Русинова.— М.: Медицина.— 1987.
2. Neurology I. Clinical Neurophysiology.— Ed. E. Stalberg, R. Young.— London.— 1981.
3. Cottrell J. E., Turndorf H. Anesthesia and Neurosurgery.— St. Louis.— 1980.
4. Recent advances in EEG and EMG data processing.— Amsterdam.— 1981.
5. Suzuki A., Ito Z., Yasui N. Topographical analysis of evoked potentials for detecting the penumbra in cerebral ischemia // In: Ischemia in Update.— Amsterdam.— 1983.— P. 191—201.
6. Hacke W. Neuromonitoring // J. Neurology.— 1985.— Vol. 232, N 3.— P. 125—140.
7. Epstein C. M., Cammon J. A., Gemmill M., Till J. Visual evoked potential pattern generation, recording and data analysis with a single microcomputer // Electroencephal. Clin. Neurophysiol.— 1983.— Vol. 56, N 3.— P. 691—693.
8. Данько С. Г., Каминский Ю. Л. Система технических средств нейрофизиологических исследований человека.— Л.: Наука, 1982.
9. Бородкин С. М., Лукьянов В. И., Зайцев В. А., Тетерина Е. А. Микрокомпьютерные средства контроля состояния мозга // Микропроцессорные средства и системы.— 1987.— № 3.— С. 67—72.
10. Guidelines in EEG and Evoked Potentials 1986 // J. Clinical Neurophysiology.— 1986.— Vol. 3.— Sup. 1.
11. Feinsod M., Selhorst J. B., Hoyt W. F., Wilson C. B. Monitoring optic nerve function during craniotomy // J. Neurosurg.— 1976.— Vol. 44, N 1.— P. 29—31.
12. Олюшин В. Е., Вайнштейн Л. Г., Меркин В. М. Устройство для динамического наблюдения зрительных вызванных потенциалов во время нейрохирургических операций // Вопросы нейрохирургии.— 1988.— № 5.— С. 45—47.
13. Вайнштейн Л. Г., Гулуева С. Х. Нарушение функционального состояния головного мозга у больных с артериальными аневризмами церебральных сосудов по данным зрительных вызванных потенциалов // В кн.: Современные вопросы функциональной нейрохирургии.— Л.: 1987.— С. 69—73.

Статья поступила 8.10.88

УДК 681.3:616.1.072

А. В. Крамаренко, Н. К. Федоров, В. Б. Шаронов

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ВВОДА И СПЕКТРАЛЬНОГО АНАЛИЗА ЭЛЕКТРОЭНЦЕФАЛОГРАММ НА БАЗЕ ПЭВМ «ИСКРА 226»

Комплекс позволяет вводить электроэнцефалограммы по восьми каналам и определять значения спектральной мощности в диапазонах δ -волн (1...4 Гц), θ -волн (4...8 Гц) и α -волн (8...13 Гц). Кроме ПЭВМ «Искра 226», укомплектованной штатным АЦП и алфавитно-цифровым печатающим устройством, комплекс включает 8-канальный электроэнцефалограф, монитор визуального контроля, 8-канальный регистратор и единственное нестандартное устройство — блок ввода сигнала в ЭВМ через один канал АЦП. Последний дает возможность использовать кабельные линии связи для дистанционной обработки электроэнцефалограмм в условиях медицинского учреждения.

Введенный сигнал подвергается спектральному анализу методом Фурье. Объем оперативной памяти, необходимой для работы программного обеспечения, составляет 58 Кбайт на языке БЕЙСИК 02.

В результате обработки на видеотерминал и АЦПУ выводятся следующие сведения: графическое отражение электрической карты функционирования мозговых структур в соответствии с расположением электродов на голове пациента;

спектральные мощности и пространственное распределение α , θ , δ -ритмов;

предварительное заключение по результатам работы алгоритма диагностики.

Для хранения данных организован архив на ГМД. На одном диске емкостью 256 Кбайт можно хранить 125 записей 4-секундного интервала 8-канальной энцефалограммы. Комплекс эксплуатируется с марта 1987 г.

310191, Харьков-191, ул. Чкалова 17, ХАИ, каф. 504; тел. 44-23-97

Сообщение поступило 5.05.88

УДК 681.3:616.1.072

Б. А. Каченко, В. Б. Шаронов, В. М. Янушков

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ РЕО И РИТМОГРАММ НА БАЗЕ ПЭВМ «ИСКРА 226»

Комплекс включает ПЭВМ «Искра 226», укомплектованную блоком БИФИ-015-14 (АЦП), ритмокардиомером РКМ-01, реоплетизмографом РПГ 2-02. Сигнал вводится через коммутатор, формирующий сигналы внешнего запуска.

Программы объемом 32 Кбайт написаны на языке БЕЙСИК 02. Оператор контролирует качество вводимого сигнала на экране дисплея и при необходимости повторяет ввод. Результаты обработки реограммы представляются в виде таблиц параметров гемодинамики; ударного объема циркулирующей крови, общенервического сопротивления и др. По результатам ритмограммы делается заключение о нарушении проводимости и ритма сердца.

Система ведет архив на ГМД. На одной стороне диска хранятся данные о 70...100 пациентах (в зависимости от объема обследования), содержащие справочные данные, значения введенных сигналов, результаты обработки, система работает с декабря 1987 г.

310191, Харьков-191, ул. Чкалова, 17, ХАИ, каф. 504; тел. 44-23-97

Сообщение поступило 5.05.88

С. В. Захаров, К. С. Марфенко, Б. И. Подлепецкий, С. В. Торубаров

СИСТЕМА НА БАЗЕ МПК БИС КР580 И КМ1813ВЕ1 ДЛЯ МОНИТОРИРОВАНИЯ ФИЗИОЛОГИЧЕСКИХ ПАРАМЕТРОВ ЧЕЛОВЕКА

Контроль функционального состояния человека в процессе его активной деятельности — актуальная задача спортивной и космической медицины. В Московском инженерно-физическом институте разработана микропроцессорная (МП) система МИФИП (МП измеритель физиологических параметров) для контроля в реальном времени частоты сердечных сокращений (ЧСС) и температуры тела человека. Система спроектирована на базе МПК БИС серии КР580 и ОЭВМ с аналоговыми устройствами ввода-вывода КМ1813ВЕ1 [1].

Особенности системы — малые габариты, высокая точность и помехоустойчивость измерений. (Основное требование, поскольку при двигательной активности человека уровень помех (артефакты движения) возрастает.) Помехи по амплитудным и частотным характеристикам близки к параметрам электрокардиографического сигнала.

Аппаратные средства

Согласно структурно-функциональной схеме система МИФИП (рис. 1) имеет четыре измерительных канала (по двум из которых измеряются ЧСС, а по двум другим — температура), вторичный преобразователь физиологических сигналов (ПФС), устройство преобразования и обработки (УПО) и четыре устройства индикации. Входные сигналы — электрокардиограмма человека (ЭКГ) и постоянное напряжение, линейно зависящее от температуры тела. Изменение напряжения (0...6 В) соответствует изменению температуры (33,0...39,0 °С). Усиливают и нормализуют измеряемые параметры предварительные усилители ЭКГ и преобразователи температура—напряжение.

Вторичный преобразователь ОЭВМ КМ1813ВЕ1 физиологических сигналов (рис. 2) предназначен для преобразования первичных аналоговых сигналов в последовательность импульсов с уровнями ТТЛ. Программа, записанная в ПФС, реализует функции преобразователей ЭКГ—ЧСС (по каналам ЧСС) и широтно-импульсных модуляторов (по каналам температуры).

Преобразователь ЭКГ—ЧСС выделяет и идентифицирует характерные зубцы ЭКГ-сигнала (обычно R-зубцы QRS-комплекса) на фоне помех [2]. Основные признаки идентификации R-зубца — частотные составляющие QRS-комплекса и его временные параметры (ширина R-зубца). Разработанный вариант преобразователя ЭКГ—ЧСС выполняет НЧ-фильтрацию четвертого порядка с частотой среза 15 Гц; ВЧ-фильтрацию второго порядка с частотой среза 8 Гц; определение максимума сигнала (пиковое детектирование); автоматическое регулирование порога срабатывания компаратора; измерение ширины R-зубцов по

уровню 0,875 от текущего значения порога; формирование сигнала на соответствующем выходе ПФС. Таким образом, на выходах ПФС, соответствующих каналам ЧСС, генерируются импульсы (200 мс), следующие с частотой сердечных сокращений. Временной промежуток между импульсами — R—R-интервал. Широтно-импульсный модулятор преобразует постоянное напряжение на входах ПФС, соответствующих каналам температуры, в последовательность импульсов (200 мс). Изменение напряжения (0...6 В) с шагом измерения 0,1 В соответствует изменению периода следования импульсов (360...1080 мс) с шагом 12 мс.

Входные сигналы ПФС имеют частотный спектр 0...100 Гц, а БИС КМ1813ВЕ1 позволяет обрабатывать сигналы в диапазоне частот от нуля до нескольких килогерц. Кроме того, 8-разрядный АЦП, размещенный на кристалле КМ1813ВЕ1, удовлетворяет требованиям по точности преобразования, необходимым для выделения R-зубцов ЭКГ-сигнала.

При проектировании программы ПФС использовались специально разработанные кросс-средства [3]. Пакет программ для разработки и отладки устройств на базе КМ1813ВЕ1 включает в себя ассемблер-дисассемблер, имитатор процессора, программу управления программатором

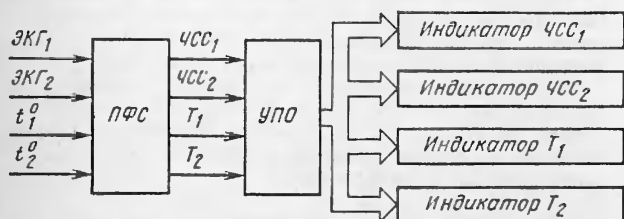


Рис. 1. Структурно-функциональная схема системы МИФИП

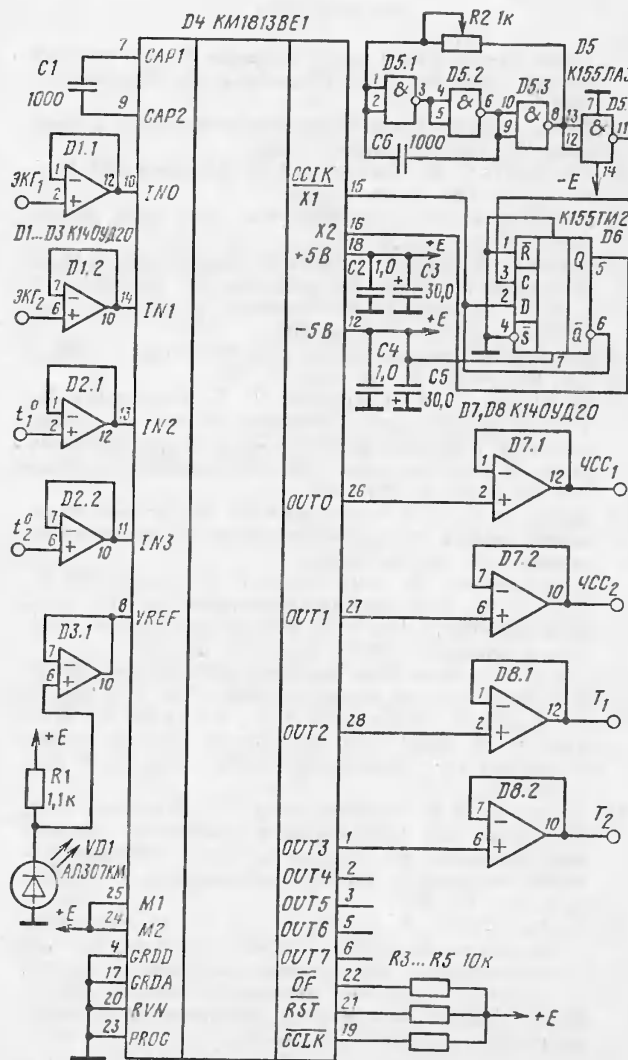


Рис. 2. Принципиальная электрическая схема ПФС


```

;Фрагмент программы ПФС. Формирование импульса,
;соответствующего R-зубцу ЭКГ. (Y6-Входная переменная)
;Период выборки 250 мкс.
;

```

```

ABS Y6 Y6 ;Выпрямление сигнала и возведение в
квадрат
LDA DAR Y6
LDA Y9 KP0
ADD Y9 Y6 R1 CND7
ADD Y9 Y6 R2 CND6
ADD Y9 Y6 R3 CND5
ADD Y9 Y6 R4 CND4
ADD Y9 Y6 R5 CND3
ADD Y9 Y6 R6 CND2
ADD Y9 Y6 R7 CND1
ADD Y9 Y6 R8 CND0
;
ABS DAR A ;Амплитудный детектор
SUB DAR Y9 L01
LDA A Y9 L01 CND5
SUB A A R13
;
LDA PRC C ;Компаратор с плавающим порогом
срабатывания
LDA DAR C
LDA C A R1 CND7
LDA C C R1 CND7
SUB C Y9
LDA DAR C
LDA C KP7
LDA C KP0 CND5
;
SUB PRC C ;Формирователь выходного импульса
LDA DAR PRC
LDA FL KM8
LDA TIM KM6 CND5
LDA DAR TIM
ADD TIM KP1 R7 CND5
LDA FL KP7 CND5
;
LDA DAR FL
;
EOP
OUT1
OUT1
OUT1

```

Рис. 3. Фрагмент программы преобразователя ЭКГ-ЧСС

(рис. 3). Конструктивно ПФС выполнен в виде платы размерами 120×80 мм, на которой размещены БИС КМ1813В41, буферные усилители, тактовый генератор, источник опорного напряжения и выходные разъемы.

Алгоритм работы УПО на базе МПК КР580 ориентирован на обеспечение наибольшей помехоустойчивости и включает в себя для каналов ЧСС усреднение длительности текущего R—R-интервала и предыдущего и исключение первого из процесса усреднения, если отклонение составило более чем ±20 %, а также выдачу в этом случае сигнала об отсутствии счета; преобразование длительности R—R-интервалов в цифровые коды, поступающие на устройства индикации ЧСС; блокировку соответствующего входа УПО в течение промежутка времени, равного 0,8 от длительности предыдущего R—R-интервала; определение допустимых пределов измеряемого параметра, устанавливаемых оператором на передней панели индикаторов ЧСС; сигнализацию о выходе значения ЧСС за верхний или нижний допустимые пределы; сброс длительности блокировки входа до минимального значения (250 мс) через каждые 30 с, а также в случаях прихода первого импульса после включения питания и прихода трех подряд R—R-интервалов, исключенных из счета (это условие необходимо для настройки на новый сердечный ритм при резкой его смене). Для каналов температуры алгоритм УПО выполняет функции, аналогичные функциям каналов ЧСС.

Программа работы УПО (1864 байт), записанная в двух ППЗУ К573РФ5, использует для работы 0,5 К ОЗУ на двух микросхемах КР541РУ1 и включает в себя основную программу и семь подпрограмм обработки прерываний. При выполнении основной программы процессор УПО большую часть времени находится в цикле, где опрашивает количество просуммированных R—R-интервалов по каждому из каналов ЧСС. Как только это количество вырастает до

шести, полученное значение усредняется и переводится в коды для выдачи на индикацию. Выход значения ЧСС за верхний и нижний допустимые пределы индицируется.

Подпрограммы обработки прерываний запускаются импульсами с выходов ПФС и таймера и измеряют R—R-интервалы и температуру, блокируют входы каналов ЧСС, определяют допустимые пределы, устанавливаемые оператором, и осуществляют мультиплексированный вывод на индикацию. УПО, выполненное в виде трех плат, соединенных друг с другом разъемными соединениями, помещено в корпус размерами 150×180×140 мм. На переднюю панель корпуса выведены одна входная розетка для подключения ПФС, четыре выходных розетки для подключения устройства индикации, вилка для подключения питания, кнопка включения питания со светодиодом индикации и кнопка «Сброс».

Информация о ЧСС и температуре индицируется с помощью градуированной светодиодной линейной шкалы (80 мм) из 64 элементов (8 корпусов АЛС362Г или АЛС362Н) и семиэлементных индикаторов АЛ304Г для цифрового значения: ЧСС — в количестве ударов в минуту, температуры — в градусах Цельсия. Цвет шкалы красный (АЛС362 Г) или зеленый (АЛС362Н), цвет семиэлементных индикаторов красный. Нижний предел измерения ЧСС — 40...80, верхний — 120...220 уд./мин с шагом установки 10 уд./мин. Для температуры нижний предел — 33,2...34,0, верхний — 37,0...39,0 °С, шаг установки — 0,2 °С. На индикаторах ЧСС расположен светодиод зеленого цвета, сигнализирующий об отсутствии свста. Устройствами индикации управляет УПО. Габаритные размеры корпуса индикаторов — 105×60×45 мм.

Технические характеристики системы МИФИП

Тактовая частота работы, МГц	
ПФС	5
УПО	1,1
Число каналов измерения	
ЧСС	2
температура	2
Диапазон измеряемых значений	
ЧСС, уд./мин	30...220
температуры, °С	33,0...39,4 °С
Абсолютная погрешность измерений	
ЧСС, уд./мин	±T1
температуры, °С	±0,1 °С
Напряжения питания, В	+5, -5, +12
Потребляемая мощность, Вт	10
Масса, кг, не более	1,5

Система передана на испытания в Институт медико-биологических проблем Минздрава СССР.

Телефон 324-31-34, Москва

ЛИТЕРАТУРА

1. Цифровой процессор обработки сигналов КМ1813ВЕ1 и его применения / Под ред. А. А. Ланнэ.— М.: ЭКОС, 1987.
2. Марфенко К. С., Подлепецкий Б. И. Микроомный детектор QRS-комплекса электрофизиологического сигнала // Тез. докл. VI Всесоюз. конф. «Измерения в медицине и их метрологическое обеспечение».— М., 1981.— С. 29.
3. Банников С. Ю., Марфенко К. С., Подлепецкий Б. И. Проектирование микроэлектронных систем на основе цифрового процессора аналоговых сигналов с использованием программной модели.— В кн.: Микропроцессорные системы и их применение.— М.: Энергоатомиздат, 1987.— С. 3—10.

Статья поступила 12.07.88

Н. Н. Щелкунов, А. П. Дианов

ОРГАНИЗАЦИЯ ОДНОКРИСТАЛЬНЫХ МИКРОКОНТРОЛЛЕРОВ

Однокристальные микроконтроллеры (ОМК) — отдельный класс микросистем (МС). Полный набор их средств расположен на одном кристалле (рис. 1). Сюда кроме центрального процессора (ЦП) входят память, подсистема ввода-вывода (ВВ), средства поддержки режима реального времени (РВ). Интеграция всех составных частей микро-ЭВМ на одном кристалле внесла ряд ограничений на принципы ее организации, потребовала новых решений в развитии ее архитектуры, не свойственных многокристальным компонентам. В результате был получен новый класс микроконтроллеров (МК) с присущими только ему принципами построения архитектуры и структурной схемы.

Однокристальные МК фирмы Intel

Период становления архитектуры 8-разрядных ОМК относят к 1976...1979 гг. [1], когда развернулась острая конкурентная борьба между тремя фирмами, выпустившими первые приборы этого класса: 8048 (Intel), 3870 (Motttek) [2, 3] и 9940 (Texas Instruments). Одни из них (3870, 9940) программно совместимы с многокристальными системами и во многом дублировали их архитектурные решения, другие (8048) имели оригинальную организацию. В процессе этой борьбы и дальнейших усовершенствований были сформированы основные принципы организации МК на одном кристалле, ставшие стандартными.

В течение четырех лет, начиная с 1976 г., фирмой Intel было разработано и получило широкое распространение целое семейство 8-разрядных ОМК iMCS-48 на основе кристалла 8048 (табл. 1). В его составе 12 микроЭВМ с единой базовой архитектурой, но различными физическими возможностями, реализованными непосредственно на кристалле, и ряд расширителей.

Базовая архитектура микросистем, построенных на основе семейства iMCS-48, включает до 4 Кбайт программного ПЗУ, реализуемого как внутренними, так и внешними по отношению к основному кристаллу средствами, до 256 байт внутренней и 256 байт внешней памяти данных, до 27 внутренних (из них три тестируемых) и 16 внешних линий ВВ, 8-разрядный таймер-счетчик, а также одноуровневую систему прерываний с двумя источниками запросов [4]. Микроконтроллер 8021 — младшая модель семейства, в которой отсутствуют средства расширения внешней памяти и система прерываний. На его основе создан простой прибор 8022 [5, 6], в котором предусмотрен 8-разрядный АЦП с двумя коммутируемыми входами. Кристалл 8041 имеет встроенный системный адаптер, обеспечивающий его подключение к шине главной системы в качестве универсального периферийного адаптера (УПА).

Набор ОМК семейства iMCS-48 и расширителей к ним был полностью освоен промышленностью к 1980 г. Сегодня приборы этого семейства рассматриваются как стандартные компоненты

для проектирования микропроцессорных средств и систем. Отечественная промышленность выпускает ряд ОМК, совместимых с семейством iMCS-48 [7, 8]: К1816ВЕ35, К1816ВЕ48, К1816ВЕ39 и К1816ВЕ49 (К1821РУ55 и К1821РЕ55 служат для расширения внешнего ОЗУ и управляющего ПЗУ соответственно).

Очередной шаг развития 8-разрядных МК — новое семейство одноразрядных микроЭВМ iMCS-51 [9]. Системы, построенные на его основе, обладают более обширным адресным пространством программ и данных, усовершенствованными средствами ВВ и поддержки режима РВ; введены команды умножения и деления, расширена группа команд над булевыми переменными. Мощное развитие получила система команд и способы доступа к отдельным элементам данных.

Сейчас в составе iMCS-51 восемь ОЭВМ с различными физическими возможностями. Расширение семейства продолжается за счет встраивания в подсистему ВВ МК специальных средств, ориентирующих МК на конкретные прикладные задачи. Так, замена в приборе 8051 последовательного адаптера общего назначения на связной адаптер с прямым доступом к внутренней памяти данных, предназначенный для организации локальных сетей, привела к появлению нового члена семейства — связного УПА 8044 [10].

Выпускается множество МК, архитектурно совместимых с семейством iMCS-51, например ОЭВМ 5000 фирмы Dallas Semiconductor [11], приборы К1816ВЕ31 и К1816ВЕ51 отечественного производства.

С появлением возможности интеграции на одном кристалле кремния более чем 100 тыс. транзисторов фирмой Intel в 1983 г. было разработано 16-разрядное семейство одноразрядных МК iMCS-96 [10]. В основе семейства — БИС 8096, 120 тыс. транзисторов которой позволили на одном кристалле разместить 16-разрядный ЦП, 8 Кбайт управляющей памяти, 232 байт памяти данных, а также мощную подсистему аналогового и цифрового ВВ с развитыми средствами поддержки режима РВ, включая скоростной вывод (СВВ). Практическое освоение перспективных 16-разрядных МК, ориентированных на рынок начала 90-х годов этого столетия, находится в начальной стадии.

Первые 4-разрядные микроконтроллерные изделия были предложены в 1971 г. и до сих пор пользуются большим спросом. Несмотря на появление 8-разрядных приборов серии К1816 4-разрядные ОМК К1814 [12, 13] и К1820 [14] продолжают широко использоваться.

К широко распространенным семействам 8-разрядных МК следует также

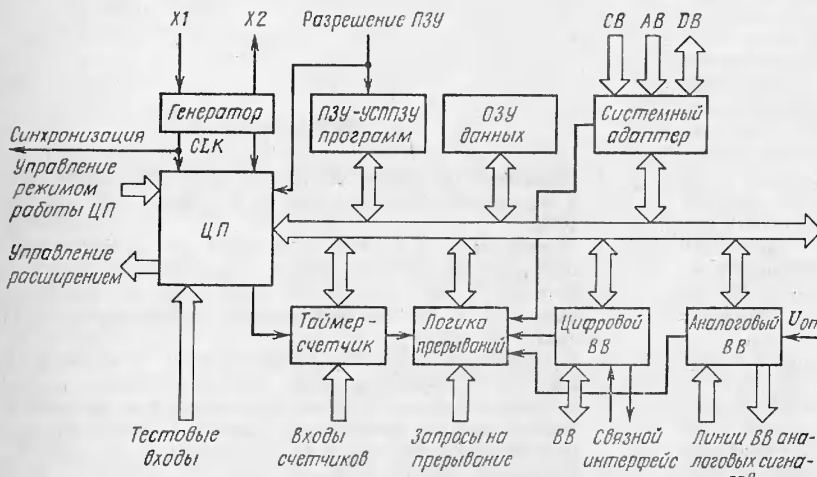


Рис. 1. Обобщенная структурная схема ОМК

Состав ОМК фирмы Intel

Тип прибора	Память программ		Память данных		Число линий		ПСА	Число таймеров	АЦП
	внутренняя, байт	внешняя, байт	внутренняя, байт	внешняя, байт	внутренних	внешних			
iMCS-48 8048 8748, K1816BE48 8035, K1816BE35 8049 8749, K1816BE49 8039, K1816BE39 8050 8040 8021 8022 8741 8742 8355 8755, K1821PE55 8155, K1821PY55 8156 8243	1К ПЗУ 1К УСППЗУ — 2К ПЗУ 2К УСППЗУ — 4К ПЗУ — 1К ПЗУ 2К ПЗУ 1К ПЗУ 2К ПЗУ — — — — —	4К 4К 4К 4К 4К 4К — — — — — — — — — — — — — —	64 64 64 128 128 128 256 256 64 64 64 128 — — — — — — — —	256 256 256 256 256 256 — — — — — — — — — — — — — — —	27 27 27 27 27 27 27 27 21 26 18 18 18 — — — — — — — — —	16 16	— —	1×8 1×8	— —
iMCS-51 8051 8751, K1816BE51 8031, K1816BE31 8052 8032 8044 8744 8344	4К ПЗУ 4К УСППЗУ — 8К ПЗУ — — 4К УСППЗУ 4К ПЗУ	64К 64К 64К 64К 64К 64К 64К 64К	128 128 128 256 256 192 192 192	64К 64К 64К 64К 64К 64К 64К 64К	32 32 32 32 32 32 32 32	— — — — — — — —	1 1 1 Сеть Сеть Сеть	2×16 2×16 — 3×16 — — 2×16 2×16 2×16	— — — — — — — —
iMCS-96 8394 8396 8094 8096 8395 8397 8095 8097	8К ПЗУ 8К ПЗУ — — 8К ПЗУ 8К ПЗУ — —	64К 64К 64К 64К 64К 64К 64К 64К	232 232 232 232 232 232 232 232	64К 64К 64К 64К 64К 64К 64К 64К	40+СВВ 40+СВВ 40+СВВ 40+СВВ 40+СВВ 40+СВВ 40+СВВ 40+СВВ	— — — — — — — —	1 1 1 1 1 1 1 1	2×16 2×16 2×16 2×16 2×16 2×16 2×16 2×16	— — — — 8×10 8×10 8×10 8×10

отнести микроЭВМ серии 6800 фирмы Motorola (6801, 6802, 6809) [15, 16], 8070 фирмы National Semiconductor [17], а также прибор Z8 фирмы Zilog [18, 19].

Микросистемы семейства MC48

Типовую систему, построенную на базе ОМК 8048, 8748 или их аналогах (рис. 2) [7...11], будем называть микросистемой 48 (MC48), соответственно MC49, MC50 и т. д.

В составе программно-доступных регистров (рис. 3) первые рабочие регистры (R0 и R1) каждого банка выполняют роль указателей данных. Ра-

бочие регистры и системный стек (16 байт) расположены во внутреннем ОЗУ данных (рис. 4). Память программ, изолированная от внутренней и внешней памяти данных, представляет два банка по 2 Кбайт, разбитых на 8 страниц каждый. В семействе систем предусмотрено изолированное пространство ВВ (рис. 5). Восьмиразрядный таймер-счетчик (рис. 6) поддерживает два режима работы, в которые он переводится командами:

START T (пуск в режиме таймера)

START CNT (пуск в режиме счетчика)

Устройство останавливает свою работу по команде

STOP TCNT (останов таймера-счетчика)

Таймер-счетчик ведет непрерывный счет импульсов на своем входе по mod 2ⁿ. При переходе через границу генерируется запрос на прерывание и устанавливается флажок переполнения TF, проверяемый командой условного перехода:

JTF addr8 (если TF=1, то PC_{0...7} ← addr8, TF ← 0)

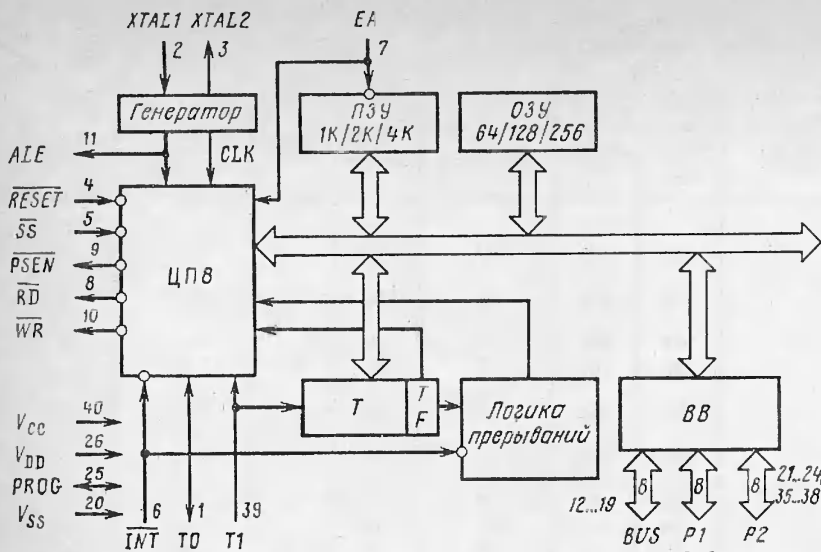


Рис. 2. Структурная схема MC48, MC49, MC50

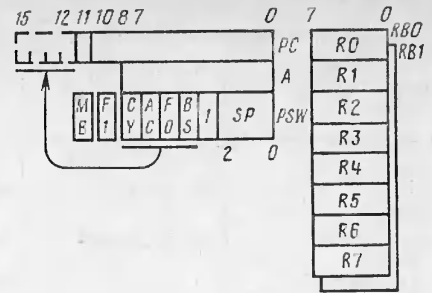


Рис. 3. Набор регистров ЦП MC48: PC — программный счетчик, А — аккумулятор, PSW — слово состояния программы, SP — указатель стека, CY — перенос, AC — дополнительный перенос; F0, F1 — флажки пользователя, BS — выбор банка рабочих регистров, ME — выбор банка памяти

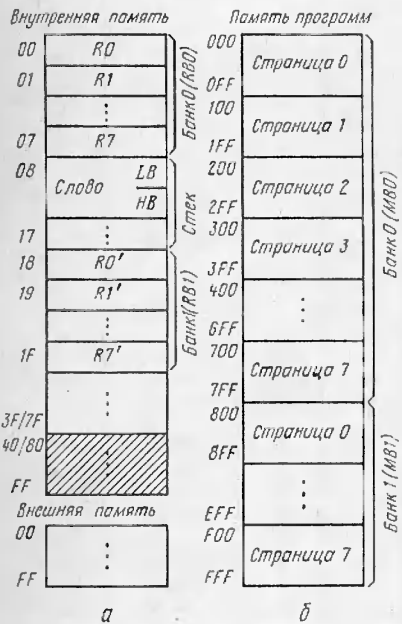


Рис. 4. Организация памяти MC48: а — данных, б — программ (4 Кбайт)

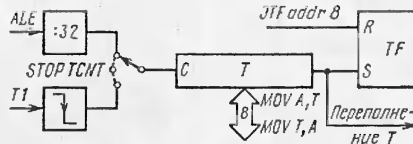


Рис. 6. Организация таймера-счетчика MC48

Одноуровневая система прерываний с двумя запросами, фиксированными векторами прерываний и приоритетами, встроенная в MC48, представлена на рис. 7.

Микросистема MC41 (рис. 8) расширена встроенным системным адаптером, удовлетворяющим требованиям шины Microbus на периферийные БИС второго поколения [20]. Средства расширения памяти программ и данных [21] удалены из системы. В состав адаптера входят двунаправленный 8-разрядный регистр слова данных DBV и 4-разрядный регистр слова состояния SW (рис. 9). MC41 как периферийное устройство взаимодействует с главным процессором через эти регистры.

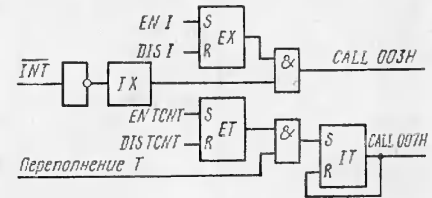


Рис. 7. Система прерываний MC48

прерывание MC41 со стартовым адресом процедуры обслуживания, равным трем (внешнее прерывание INT для MC48). Информация из главной системы в DBV вводится как при A0=0, так и при A0=1 (состояние A0 фиксируется флажком F1). Обращение по адресу A0=0 резервируется для передачи непосредственно данных, по адресу A0=1 — для передачи управляющих слов. Со стороны шины Microbus флажок F0 доступен только для чтения в составе SW.

Для процессора MC41 буфер DBV заменяет порт BUS микросистемы MC48 (все флажки SW программно-доступны). Флажки F0, F1 — одноименные флажки пользователя MC48 (IBF и OBF — два новых тестируемых флажка).

Набор команд MC41 совместим с системой команд MC48 на уровне объек-

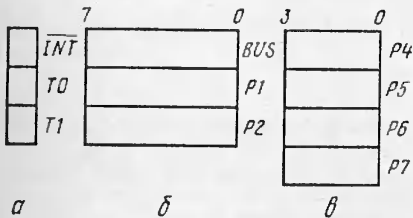


Рис. 5. Пространство ВВ MC48: а — тестируемые флажки, б — внутренние и в — внешние порты ВВ

Операции доступа к MC41				
A0	RD	WR	CS	Операция
0	0	1	0	D ← DBV, OBF ← 0
1	0	1	0	D ← SW
0	1	0	0	DBV ← D, F1 ← 0, IBF ← 1
1	1	0	0	DBV ← D, F1 ← 1, IBF ← 1
X	1	1	0	Нет операции
X	X	X	1	Нет операции

Ввод-вывод данных сопровождается аппаратным сбросом-установкой флажков готовности OBF/IBF. Установка IBF вызывает генерацию запроса на

ного кода, однако в MC41 запрещены следующие команды: обращения к отсутствующей в системе внешней памяти данных XSEG

MOVX	A, @Ri	(A ← XSEG(Ri), i=0...1)
MOVX	Ri@A	(XSEG(Ri) ← A, i=0...1)

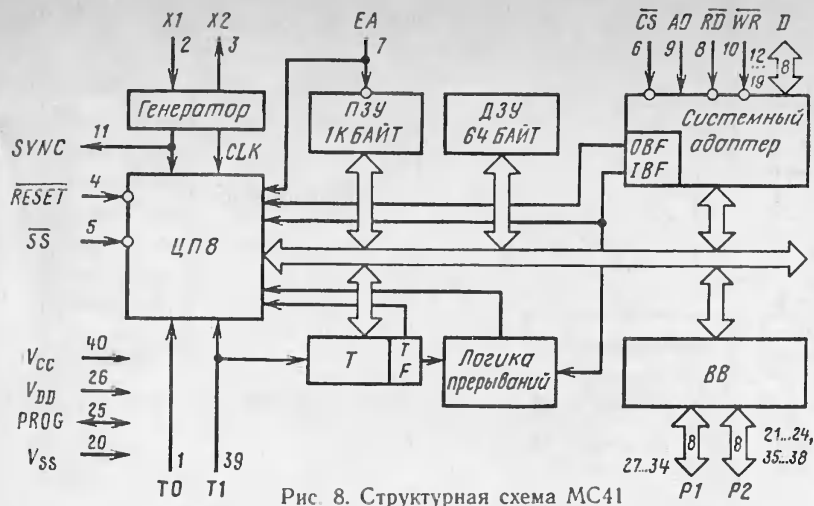


Рис. 8. Структурная схема MC41

выбора банков памяти программ (не реализована возможность ее внешнего расширения)

что позволяет организовать программное тестирование флажков IBF и OBF состояния буфера данных DBB. Такая проверка необходима при построении программно-управляемого ВВ. Команды ввода-вывода

SEL MB0 (MB←0)
SEL MB1 (MB←1)

IN OUT A, DBB DBB, A (A←DBB, IBF←0) (DBB←A, OBF←1)

тестирования входа \overline{INT} (выполняет функцию запроса на прерывание по флагу IBF)

связывают буфер DBB с аккумулятором, обеспечивая программный доступ к данным со стороны процессора MC41

JNI addr8 (если INT=0, то PC_{0..7}←addr8)

обслуживания порта BUS (используемого для доступа к буферу DBB)

(поддерживают согласованное управление флажками IBF и OBF, отражаю-

INS A, BUS (A←BUS)
OVTL BUS, A (BUS←A)
ANL BUS, # data (BUS←BUS and data)
ORL BUS, # data (BUS←BUS or data)

разрешения выдачи CLK на вывод T0

ENT0 CLK

Вместе с этими введены четыре новые команды (табл. 2), а команды условного перехода внутри текущей страницы (рис. 4) расширены:

щими состояние DBB). Флажок IBF сбрасывается при чтении буфера, OBF устанавливается при записи. Внешнее прерывание \overline{INT} микросистемы MC48 в MC41 используется для принятия запроса на обслуживание по флажку IBF, поэтому команды разрешения (ENI) и запрета (DIS I) прерывания получают другую интерпретацию. Байт информации может быть введен

JNIBF addr8 (если IBF=0, то PC_{0..7}←addr8)
JOBF addr8 (если OBF=1, то PC_{0..7}←addr8)

Таблица 2
Команды обслуживания DBB

Команда	Код	Описание
JNIBF addr8	D6	Если IBF=0, то PC _{0..7} ←addr8
JOBF addr8	86	Если OBF=1, то PC _{0..7} ←addr8
IN A, DBB	22	A←DBB, IBF←0
OUT DBB, A	02	DBB←A, OBF←1

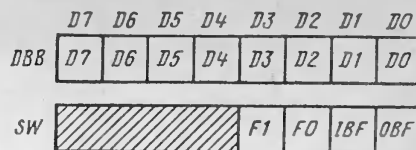


Рис. 9. Регистры системного адаптера MC41

в буфер DBB главным ЦП как байт данных или как командный байт в зависимости от адреса, по которому ведется запись.

Со стороны процессора MC41 тип информации кодируется сбросом-установкой флажка F1. Флажок IBF определяет состояние буфера. Если буфер заполнен, то флажок F1 показывает вид информации в нем.

Главный ЦП может вводит информацию в DBB в любое время, тогда как процессор МК записывает информацию в DBB, только если ему дан запрос от главного ЦП. Информация обычно передается в виде пакетов, состоящих из последовательных байтов данных и за-

вершающихся командным байтом с нулевым состоянием, указывающим на конец передачи. Первый байт пакета может содержать число байтов в пакете. Возможны и другие протоколы взаимодействия главного ЦП с MC41.

Микросистемы семейства MC51

Организация однокристалльных 8-разрядных MC51 и MC52 (рис. 10) одинакова за исключением объемов внутренних ПЗУ и ОЗУ, а также числа 16-разрядных таймеров счетчиков (табл. 1). Набор их программно-доступных регистров (рис. 11) — расширение регистровой памяти MC48 (рис. 3). Обе архитектуры относятся к классу аккумуляторных с переключаемыми банками рабочих регистров. В состав регистрового набора введены два новых элемента: расширение аккумулятора В и 16-разрядный указатель данных DPTR, образованный из двух байтовых регистров DPH и DPL. Среди новых флажков следует отметить флажки арифметического переполнения OV и четности P. Указатель стека SP организован в виде отдельного 8-разрядного регистра.

В MC51 предусмотрены четыре банка по восемь рабочих регистров R0..R7; банк выбирается полем RS в слове состояния программы. Они выполняют общеселевые функции по промежуточному хранению данных. По аналогии с MC48 R0 и R1 каждого банка реализуют также функцию 8-разрядных указателей данных. С помощью набора рабочих регистров существенно уменьшается длительность переключения кон-

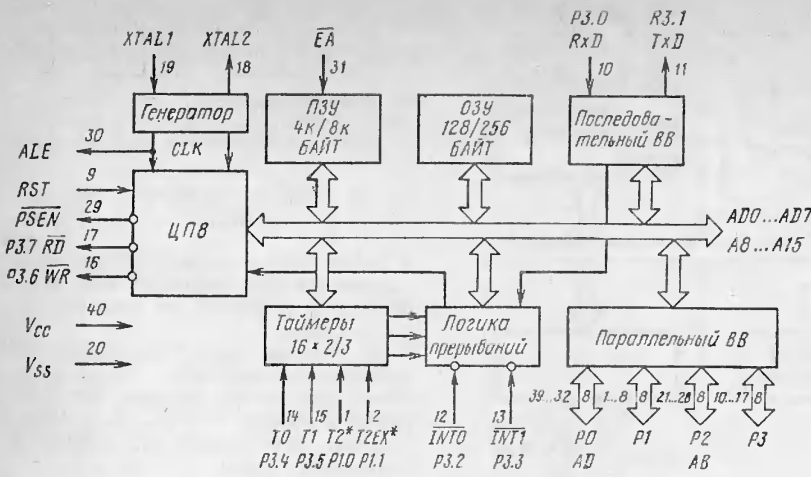


Рис. 10. Структурная схема MC51, MC52

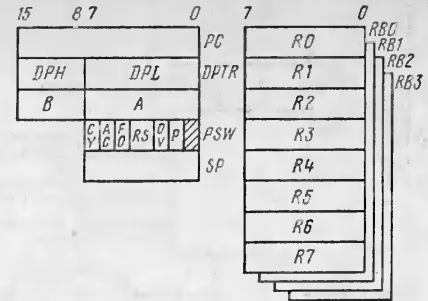


Рис. 11. Набор регистров ЦП MC51

текстов ЦП, что очень важно для MC реального времени. В MC51 отсутствуют ограничения, накладываемые на процедуры обслуживания прерываний, свойственные MC48.

Память программ (64 Кбайт — однородная линейная область, реализуемая как внутренними, так и внешними средствами. Для совместимости с MC48 предусмотрен ряд команд, которые позволяют рассматривать память программ в виде набора 2-Кбайтных банок (рис. 4). Подобно архитектуре MC48 все банки рабочих регистров, а также системный стек располагаются во внутренней памяти данных. Определены два способа адресации памяти: прямой (direct) и косвенный (@Ri, где i = 0...1). С помощью прямой адресации доступна только младшая часть адресного пространства внутренней памяти данных (128 байт), тогда как косвенная обеспечивает доступ к любой ее ячейке из диапазона 256 байт. Введение прямой адресации расширило возможности ОМК по обработке данных. В частности, появились средства доступа к рабочим регистрам и системному стеку, интерпретируемым как обычные ячейки памяти.

Микроконтроллер MC51 имеет развитую подсистему ВВ и средства управления режимом ПВ. Для их управления в МК предусмотрен ряд регистров (рис. 12), которые размещены в отдельном прямо адресуемом пространстве специальных регистров (128 байт). Сюда же включены и некоторые регистры ЦП. Пространство специальных регистров вместе с младшей частью адресного пространства внутренней памяти данных образуют прямо адресуемую область (рис. 13). При этом сначала размещается младшая половина пространства внутренней памяти данных, а затем пространство специальных регистров.

Центральный процессор MC51 содержит специальную логику для выполнения ряда однобитовых операций, в которых роль аккумулятора реализует флажок переноса CY. Для хранения булевых данных в архитектуре MC51

7	6	5	4	3	2	1	0	
87	86	85	84	83	82	81	80	P0
						T2EX*	T2*	P1
97	96	95	94	93	92	91	90	P2
A7	A6	A5	A4	A3	A2	A1	A0	P3
RD	WR	T1	T0	INT1	INT0	TxD	RxD	BO
B7	B6	B5	B4	B3	B2	B1	B0	
								SBUF
								99
								SCON
								98
15								
								8
								T0
								T1
								TMOD
								89
								TCON
								88
								T2*
								RCAP2*
								87
								T2CON*
								88
								IE
								88
								IP
								88
								PCON
								87

Рис. 12. Специальные регистры MC51, MC52

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	RO	R1	R2	R3	R4	R5	R6	R7	RO'	R1'	R2'	R3'	R4'	R5'	R6'	R7'
1	RO''	R1''	R2''	R3''	R4''	R5''	R6''	R7''	RO'''	R1'''	R2'''	R3'''	R4'''	R5'''	R6'''	R7'''
2	00...07	08...0F	10...17	18...1F	20...27	28...2F	30...37	38...3F	40...47	48...4F	50...57	58...5F	60...67	68...6F	70...77	78...7F
3																
4																
5																
6																
7																

а

8	PD 80...87	SP	DPL	DPH					PCON	TCON 88...8F	TMOD	TL0	TL1	TH0	TH1	
9	P1								SCON 98...9F	SBUF						
A	P2								IE AB...AF							
B	P3 30...37								IP 88...8F							
C	* C0...C7								T2* CON CB...CF		RC* AP2L	RC* AP2H	TL2*	TH2*		
D	PSW 90...97								DB...DF							
E	ACC E0...E7								EB...EF							
F	8 F0...F7								FB...FF							

б

Рис. 13. Прямо адресуемая часть памяти:
а — внутренних данных, б — специальных регистров

предусмотрено отдельное прямо адресуемое пространство BSEG (256 бит), которое физически совмещено с прямо адресуемой частью внутренней памяти данных и областью специальных регистров. На рис. 13 адреса этих битов указаны внутри байтов, совмещенных с данным пространством. Например, аккумулятор А, доступный как элемент регистровой памяти, может быть прямо адресован и как элемент пространства специальных регистров (адрес 0E0H), и как область пространства битов с адресами 0E0H...0E7H. При этом битовый адрес 0E0H относится к младшему разряду аккумулятора. Косвенная адресация байта с адресом 0E0H обеспечивает доступ к ячейке в старшей половине адресного пространства внутренней памяти данных (имеет смысл только для MC52).

Размещенные в области специальных регистров (рис. 12) четыре 8-разрядных псевдодвунаправленных порта ВВ совмещены с битовым пространством, что обеспечивает доступ к отдельным

их разрядам независимо друг от друга. Например, благодаря командам условного перехода типа

JB bit, rel (если BSEG(bit)=1, то PC←PC+rel)

каждая линия портов ВВ тестируется отдельно.

Расширение области ВВ может быть выполнено за счет внешнего пространства памяти данных. При обращении к внешней памяти порты P0, P2 выполняют роль системной магистрали AD и АВ соответственно. Линии порта P3 реализуют процесс управления циклами обмена и другие специальные функции аппаратного уровня (см. рис. 10).

В состав MC51 введен дуплексный периферийный связной адаптер (ПСА), который может быть запрограммирован для работы в одном из четырех основных режимов:

0 или синхронный последовательный ВВ со скоростью OSC/12;

1 или асинхронный с 10-битовым кадром и переменной скоростью передачи;

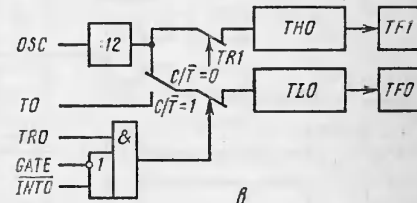
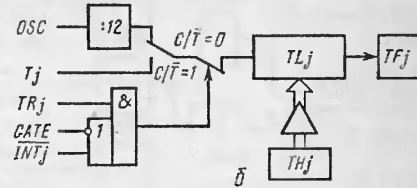
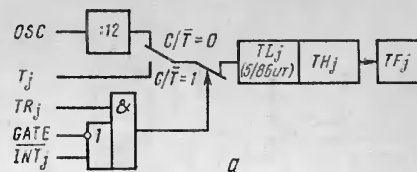


Рис. 14. Организация таймеров-счетчиков MC51:
а — режим 0 и 1, б — режим 2, в — режим 3

2 или асинхронный с 11-битовым кадром и фиксированной скоростью передачи: OSC/32 или OSC/64;

3 или асинхронный с 11-битовым кадром и переменной скоростью передачи. Входные и выходные данные хранятся в буферном регистре SBUF с адресом 99H. Приемопередатчики управляются с помощью слова управления и состояния SCON.

Архитектура MC51 предоставляет пользователю два 16-разрядных таймера-счетчика, состояние которых отражается в программно-доступных регистровых парах (TH0, TL0), (TH1, TL1).

Еще один таймер-счетчик добавляется в MC52 (TH2, TL2). В режиме таймера считаются синхрипульсы, следующие с частотой OSC/12, в режиме счетчика отслеживаются перепады из состояния с высоким уровнем напряжения в низкое на входах T0, T1 и T2 соответственно. Регистры режима TMOD и управления TCON управляют первыми двумя таймерами-счетчиками. Каждое устройство (рис. 14) имеет четыре подрежима, один из которых аналогичен режиму работы таймера-счетчика MC48. Для управления работой третьего таймера-счетчика в MC52 предусмотрен регистр T2CON и 16-разрядный регистр захвата-автозагрузки RCAP2. Функциональные свойства устройства значительно шире, чем у первых двух.

Архитектура MC51 поддерживает

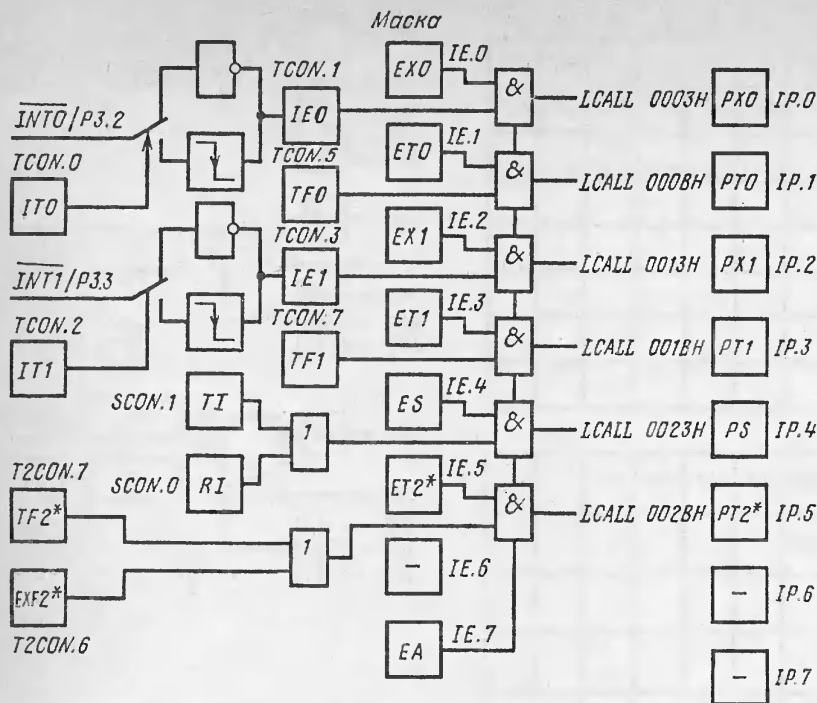


Рис. 15. Система прерываний MC51, MC52

2-уровневую приоритетную систему прерываний с пятью (шестью в случае MC52) источниками запросов, имеющих фиксированные вектора прерываний (рис. 15). Регистр приоритетов IP связывает с каждым источником приоритет (0 или 1). Один из разрядов IE маскирует запросы. Внешние линии INT0 и INT1, которые могут быть запрограммированы на срабатывание по уровню и спаду сигнала, флажки переполнения таймеров-счетчиков TF0, TF1 и TF2, запросы на обслуживание ПСА TI и RI, а также флажок EXF2 в T2CON служат источниками прерываний. Флажки внутренних запросов TF0 и TF1 сбрасываются автоматически при вызове процедуры обслуживания прерываний. Состояние TI, RI, TF2 и EXF2 не изменяется, они используются для уточнения источника запроса по методу полинга. Все флажки, генерирующие запросы на прерывания, могут быть восстановлены программно, что дает возможность реализовать вызов соответствующих процедур обслуживания непосредственно из программ. Входы INT0, INT1 дублированы программно-управляемыми флажками IE0 и IE1 из регистра TCON.

ОМК специального назначения имеют архитектуру MC51. Например, специализация МК 8744 выполняется за счет подсистемы ВВ. Для управления новыми аппаратными средствами в архитектуру MC51 вводится ряд новых регистров, которые располагаются на свободных местах пространства специаль-

ных регистров. При этом базовая архитектура и система команд не изменяются. Такая гибкость архитектуры MC51 обеспечила приборам ее семейства перспективность и совместимость в новых разрабатываемых.

Телефон 408-62-22, Москва

ЛИТЕРАТУРА

1. Альтман Л. Расширение семейств микрокомпьютеров. Часть I: Новые приборы // Электроника.— 1977.— № 25.— С. 23—38.
2. Дозье Х., Грин Р. Однокристалльный микрокомпьютер с большой оперативной и постоянной памятью // Электроника.— 1978.— № 10.— С. 26—34.
3. Герсон С. Микрокомпьютер со сменным стираемым ППЗУ — эффективное средство эмуляции однокристалльных микроконтроллеров // Электроника.— 1980.— № 3.— С. 34—40.
4. Уокерли Дж. Архитектура и программирование микроЭВМ: В 2-х кн. Пер. с англ.— М.: Мир, 1984.— Кн. 2, 341 с.
5. Монолитный микрокомпьютер с аналогово-цифровым преобразователем / В. Чек, Е. Чжен, Г. Хилл, М. Холлен. Дж. Миллер // Электроника.— 1978.— № 11.— С. 36—44.
6. Иттнер В. Ф., Миллер Дж. А. Новые функциональные возможности

однокристалльного микрокомпьютера // Электроника.— 1978.— № 15.— С. 48—54.

7. Кобылинский А. В., Липовский Г. П. Однокристалльные микроЭВМ серии K1816 // Микропроцессорные средства и системы.— 1986.— № 1.— С. 10—19.
8. Каган Б. М., Сташин В. В. Основы проектирования микропроцессорных устройств автоматики.— М.: Энергоатомиздат, 1987.— 304 с.
9. Мун Дж. Микрокомпьютер для эмуляции // Электроника.— 1980.— № 16.— С. 48—53.
10. Microcontroller Handbook. Intel, 1985.
11. Коул Б. Перепрограммируемый микроконтроллер — мечта разработчика // Электроника.— 1987.— № 6.— С. 19—25.
12. Однокристалльные 4-разрядные микроЭВМ серии K1814 / В. Н. Златопольский, Лобов И. Е., Стоянов А. И., Шадрин И. А. // Микропроцессорные средства и системы.— 1985.— № 1.— С. 3—10.
13. Крылов Е. И. Однокристалльные микроЭВМ серий K1814, K1820, K1816 // Микропроцессорные средства и системы.— 1985.— № 2.— С. 3—7.
14. Однокристалльные 4-разрядные микроЭВМ серии KP1820 / В. А. Бобков, Ю. Н. Бурмистров, В. А. Кособрухов, Ю. В. Уткин, Б. Н. Чернуха // Микропроцессорные средства и системы.— 1986.— № 1.— С. 19—27.
15. Обеспечение совместимости — надежный путь расширения микрокомпьютерного семейства / М. Уайлз, Ф. Мьюза, Т. Ф. Риттер, Боуни, Т. Гантер // Электроника.— 1978.— № 3.— С. 23—26.
16. Смит Д. У. Микрокомпьютер, работающий автономно или в комплексе с другими приборами // Электроника.— 1979.— № 25.— С. 41—51.
17. Хьюз Ф., Дьюгэн Т., Хэнди Дж. Однокристалльный микрокомпьютер, предназначенный для многопроцессорной работы // Электроника.— 1980.— № 22.— С. 66—71.
18. Пьюто Б. Л., Просенко Г. Дж. Однокристалльный микрокомпьютер Z8. Часть 1: Широкие аппаратные возможности // Электроника.— 1978.— № 18.— С. 42—49.
19. Басс С., Эстрин Дж., Пьюто Б. Л. Однокристалльный микрокомпьютер Z8. Часть 2: Гибкие программные средства // Электроника.— 1978.— № 18.— С. 49—55.
20. Форс. Стандартная микропроцессорная шина, упрощающая задачи разработки микрокомпьютеров // Электроника.— 1978.— № 15.— С. 33—41.
21. UP1-41 User's manual. Intel, 1977.

Статья поступила 6.11.87

Ю. А. Орестов, В. Н. Бобылев

ПРОГРАММИРУЕМЫЙ МИКРОКОНТРОЛЛЕР НА ОСНОВЕ ОЭВМ КР1816ВЕ31

Программируемый контроллер — это микропроцессорное устройство, которое принимает входные сигналы, обрабатывает их и в соответствии с программой, заложенной в его память, генерирует сигналы управления. Число входов-выходов, являющееся одной из важнейших технических характеристик контроллера, может достигать сотен и даже тысяч. Контроллеры, имеющие не более 32 входов-выходов, называются микроконтроллерами и используются для управления несложными объектами или их частями.

Увеличение доли микроконтроллеров в общем объеме выпуска программируемых контроллеров объясняется повышенной надежностью систем управления на их основе, простотой диагностики и приемлемой ценой. Дальнейшее усовершенствование микроконтроллеров связано с применением более совершенных микропроцессоров, устройств памяти и интерфейсных схем.

Для большинства современных микроконтроллеров характерно использование 8-разрядных МП серий КР1816 и КР580.

Задача создания эффективных микроконтроллеров значительно упростилась с появлением ОЭВМ КР1816ВЕ31. Ее состав:

8-разрядный центральный процессорный элемент;

логический процессор с поразрядной, побитовой и побайтной обработкой данных;

встроенный тактовый генератор;

встроенное ЗУПВ данных (128 байт);

20 регистров специального назначения;

восемь шин ввода-вывода;

два встроенных 16-разрядных таймера-счетчика;

двусторонний порт последовательного ввода-вывода;

система прерываний двумя уровнями приоритета от двух внешних и трех внутренних источников;

адресное пространство (64К) для внешнего ЗУПВ данных;

адресное пространство (64К) для внешней программной памяти.

По всем основным характеристикам ОЭВМ КР1816ВЕ31 превосходит предшествующие модели МП и ОЭВМ, уступаая МП КР580ВМ80А лишь по глубине стека (см. таблицу).

Эффективная система команд позволяет разрабатывать программы, занимающие в 2—4 раза меньший объем памяти и выполняющиеся значительно быстрее. Это объясняется тем, что возможность побитовой и побитовой обработки данных, разветвленная система адресации, большое число внутренних регистров и встроенное ОЗУ данных позволяют обойтись при выполнении задач пользователя меньшим числом операций с данными, а практически все команды выполняются за один или два машинных цикла. Например, при максимальной тактовой частоте 12 МГц из 111 команд 64 осуществляются за 1 мкс, 45 — за 2 мкс и лишь две (умножение и деление) — за 4 мкс.

На базе ОЭВМ КР1816ВЕ31 разработан программируемый микроконтроллер (рис. 1). Структурно он выполнен на плате Е2 размерами 233,3×220 мм, снабженной двумя разъемами типа СНП59. Каждый из разъемов имеет по 96 контактов.

На один из разъемов выведена системная шина контроллера, на другой — программируемые входы-выходы, последовательные и параллельные интерфейсы связи.

Сравнительные характеристики ОЭВМ с фиксированными системами команд

Тип МП	КР580ВМ80А	КР1816ВЕ48	КР1816ВЕ31
Разрядность, бит	8	8	8
Тактовая частота, МГц	2,5	6	12
Объем, Кбайт:			
внешней адресуемой памяти,	64	4	64
программ данных	64	Нет	64
внутренней памяти данных	Нет	64	128
Число РОН	6	16	20
Число прерываний:			
внутренних	Нет	1	3
внешних	1	1	2
Встроенные программируемые интерфейсы	Нет	Нет	Последовательный
Встроенные таймеры	Нет	1	2
Число команд	78	96	111
Глубина стека, Кбайт	64	0,016	0,128
Число источников электропитания	3	1	1
Возможность поразрядной обработки данных	Нет	Нет	Есть

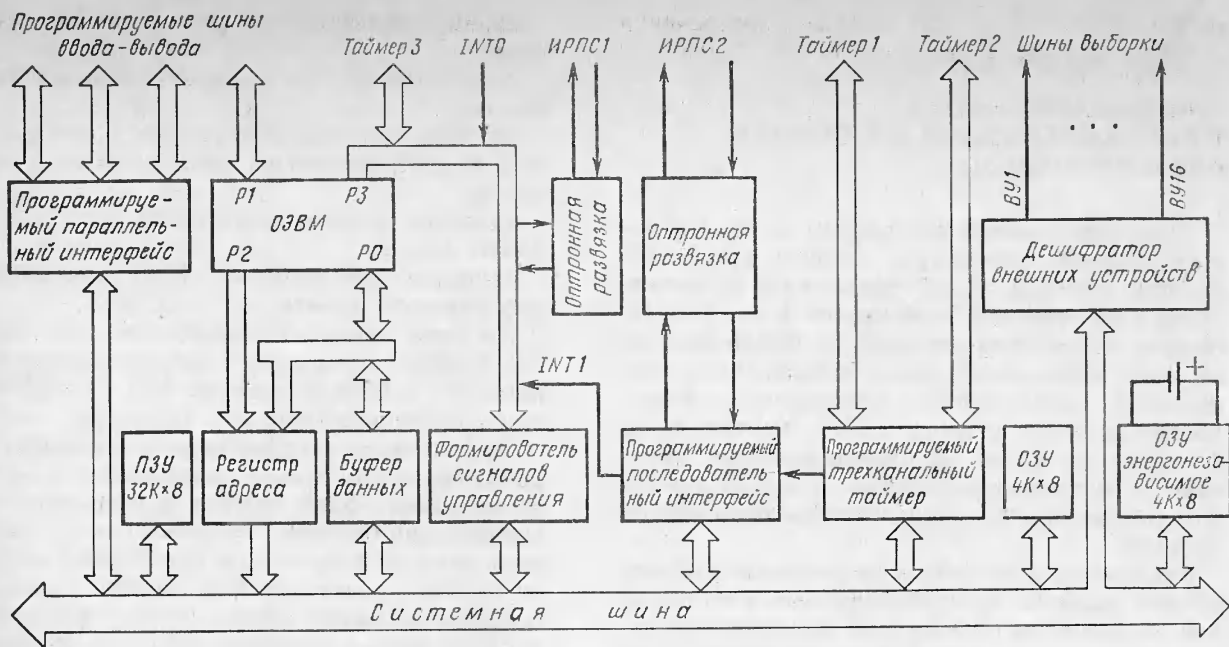


Рис. 1. Структурная схема программируемого микроконтроллера

Контроллер может применяться автономно или дополняться периферийными модулями, подключаемыми к системной шине.

На плате контроллера размещены:

ОЗВМ КР1816ВЕ31 с дополнительными буферными элементами и внешним тактовым генератором с частотой 8 МГц;

адресный дешифратор, выполненный на программируемой логической ИС КР556РТ2;

колотки с элементами коммутации, обеспечивающие размещение в них БИС РПЗУ серии К573 (К573РФ2, К573РФ5, К573РФ41, К573РФ42 или К573РФ4) с объемом памяти программ 8...32К;

ОЗУ (4К×8 бит);

энергонезависимое ОЗУ (4К×8 бит), выполненное на БИС К537РУ8А, с временем хранения информации после отключения электропитания не менее 100 ч;

устройство оптронной развязки и формирования выходного сигнала для последовательного порта ввода-вывода информации ОЗВМ КР1816ВЕ31;

устройство последовательного ввода-вывода информации с оптронной развязкой и формированием выходного сигнала, выполненное на БИС КР580ИК51;

программируемый таймер КР580ВИ53, один из каналов которого используется как делитель частоты для устройства последовательного ввода-вывода, а остальные — как дополнительные таймеры-счетчики;

программируемый параллельный интерфейс КР580ВВ55А, который вместе с восемью шина-

ми ввода-вывода ОЗВМ КР1816ВЕ31 обеспечивает дискретной информацией 32 шины ввода-вывода.

Для настройки аппаратной части контроллера, разработки и отладки ПО предоставляется автономное отладочное устройство (рис. 2). Оно обеспечивает:

внутрисхемную эмуляцию ОЗВМ КР1816ВЕ31;

ввод отлаживаемой программы в машинном коде с помощью 16-ричной клавиатуры и текущего адреса, данных и другой служебной информации на дисплей;

индикацию по вызову оператора содержимого внутренних регистров ОЗВМ и его модификацию (при необходимости);

ввод адресов контрольных точек (точек прерывания эмуляции);

выполнение введенной программы пользователя в реальном времени и с прерыванием эмуляции;

прерывание эмуляции после выполнения каждой команды или по контрольным точкам;

обмен информацией с внешними устройствами (ПЭВМ, магнитофоном и др.);

контроль и настройку аппаратных средств контроллера методом тестирования статическими сигналами.

Указанные функции отладочного устройства реализуются с помощью резидентной управляющей программы монитора, которая включает в себя подпрограммы обслуживания индикации, клавиатуры, режима загрузки; служебные подпрограммы вывода, модификации и загрузки внутренних регистров и ячеек встроен-

ного ОЗУ данных ОЭВМ, а также обеспечения режимов прерывания эмуляции и сервисных подпрограмм.

Как и любое отладочное средство, разработанное устройство накладывает ряд ограничений на отлаживаемые с его помощью контроллеры: вход немаскируемого прерывания INT не используется, адресные пространства памяти программ и данных совмещены и ограничены объемом 32 Кбайт. Однако эти ограничения не очень жесткие, так как немаскируемое прерывание обычно применяется для обслуживания аварийных ситуаций, реализация и отладка которых осуществляются после отработки основного ПО, а выделяемый пользователю объем памяти вполне достаточен для широкого круга задач.

Указанные ограничения компенсируются простотой разработанного отладочного устройства, его низкой стоимостью, возможностью работы в цеховых условиях без применения дополнительного оборудования.

Основа отладочного устройства — ОЭВМ K1816BE31, обеспечивающая попеременное выполнение программы монитора, защитой в ПЗУ, и программы пользователя, хранящейся в ОЗУ

программ пользователя или в памяти микроконтроллера. Для эмуляции все выходы ОЭВМ, кроме INT0 и шины данных, выводятся на разъем непосредственно, а шина данных — через управляемый буфер. Эмулирующая ОЭВМ дополнена регистром адреса, буфером данных, формирователем сигналов управления внутренней системной шины.

К системной шине подключаются узлы, обеспечивающие функционирование устройства: клавиатура; индикаторная панель; управление прерыванием эмуляции; ОЗУ программ пользователя, контрольных точек и регистров; программируемый интерфейс ввода-вывода.

Клавиатура содержит 23 клавиши, в том числе 16 символьных и семь функциональных. Подавление дребезга и идентификация клавиши осуществляются программно. Панель индикации — шесть семисегментных светодиодных индикаторов, с помощью которых выводятся текущий адрес и данные, а также служебная информация, позволяющая пользователю проводить тестирование аппаратных средств или отладку программ.

Узел управления прерыванием эмуляции гарантирует определение типа выполняемой

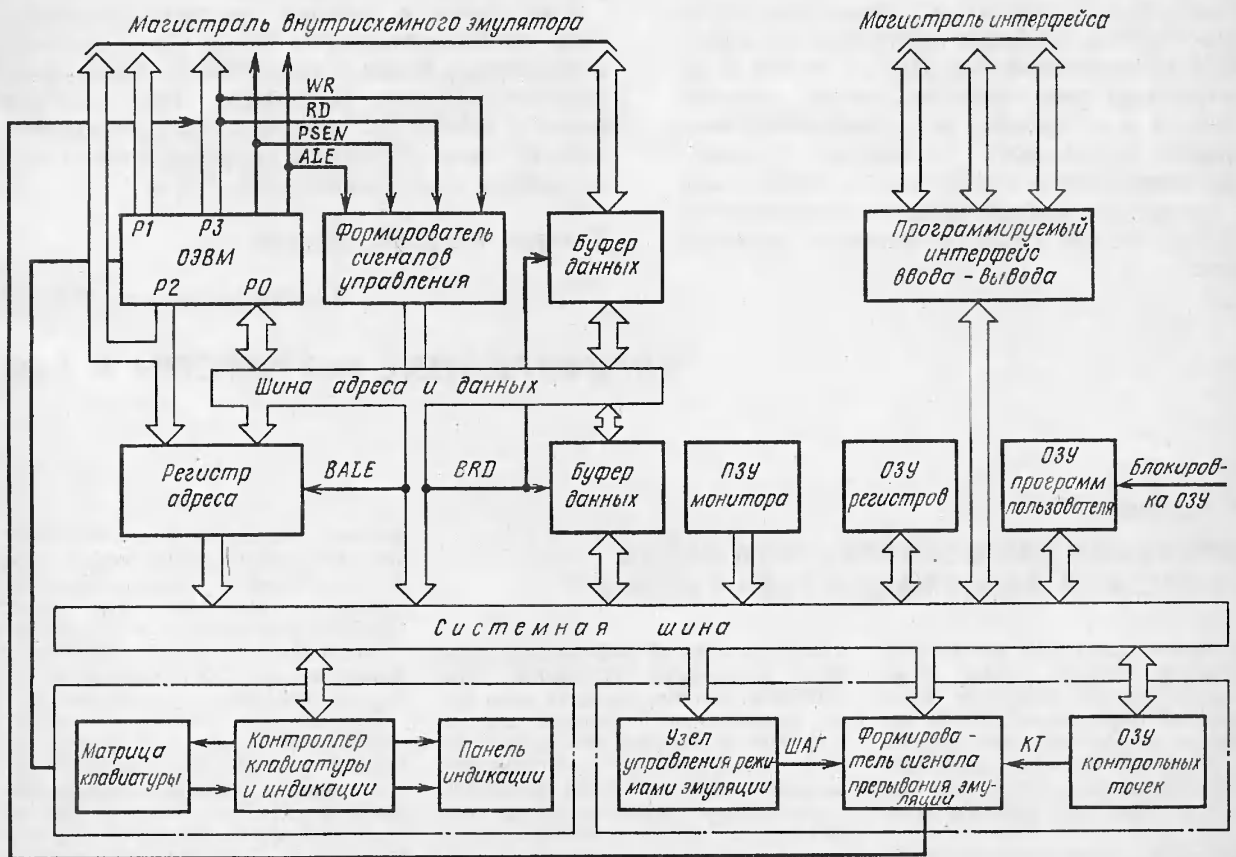


Рис. 2. Структурная схема отладочного устройства

команды, подсчет числа тактов и в нужный момент времени формирует сигнал INTO, прерывающий процесс эмуляции.

ОЗУ программ пользователя (4 Кбайт) имеет принудительную блокировку доступа и может перемещаться в адресном пространстве, создавая различные конфигурации памяти. Совместная работа с реальной памятью контроллера позволяет наращивать объем отлаживаемых программ. Программы пользователя вводятся с клавиатуры отладочного устройства или от внешнего источника с использованием программируемого интерфейса ввода-вывода.

В ОЗУ контрольных точек (4 Кбит) хранятся признаки прерывания эмуляции. Контрольные точки заносятся с клавиатуры пользователем. В ОЗУ регистров хранится содержимое внутренних регистров и ячеек ОЗУ ОЭВМ (после каждого прерывания эмуляции перезаписывается). При необходимости содержимое отдельных ячеек может быть просмотрено и модифицировано пользователем, что позволяет ему корректировать ход выполнения программ, так как перед выполнением следующей после прерывания эмуляции команды содержимое ОЗУ регистров переписывается во внутренние регистры и ячейки ОЗУ ОЭВМ.

С помощью 24 шин ввода-вывода дуплексные каналы байтового обмена подключаются к микроЭВМ «Электроника 60», ДВК, СМ1800 и др.

Разработан ряд сервисных узлов, подключающихся к интерфейсу и расширяющих возможности отладочного устройства, например оперативная память с сохранением информации при отключении электропитания, программатор ЭСРПЗУ, модем связи с бытовым магнитофоном.

Для отладки аппаратных средств применяется специальный режим работы интерфейса; его шины используются для эмуляции основных сигналов ОЭВМ, в том числе обязательно сигналов портов P0 и P2, ALE, PSEN, WR, RD. С помощью специального ПО обеспечивается статическая эмуляция машинных циклов «Ввод команды», «Ввод» и «Вывод». Такой прием оказывается весьма эффективным, так как ОЭВМ КР1816ВЕ31 — динамическая система, которую не удастся зафиксировать в каком-либо статическом состоянии. Глубина контроля отлаживаемого устройства определяется пользователем, именно он разрабатывает программу тестирования.

Для физической реализации режима в составе отладочного устройства есть специальный кабель с вилкой, которая вставляется в колодку ОЭВМ контроллера.

После проведения отладки в статическом режиме можно переходить к дальнейшей проверке в режиме внутрисхемной эмуляции с использованием библиотеки сервисных программ тестирования типовых устройств: памяти, программируемых интерфейсов. Сервисные программы выполняются по командам или в режиме реального времени.

Разработка и отладка программ пользователя небольшого объема может производиться в машинных кодах с применением предложенного отладочного устройства. При объемах более 4 Кбайт ПО рекомендуется разрабатывать на кросс-средствах, например отладочном комплексе «Электроника МС 0701».

Телефон 460-41-66, Москва

Статья поступила 15.04.88

ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА И УСО

УДК 681.326—181.4

О. С. Андрушин

СОПРЯЖЕНИЕ ДВК С УСТРОЙСТВОМ ВВОДА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ ТИПА СМП6410

Последние модели ДВК хотя и имеют графические дисплеи, однако в их составе отсутствуют устройства ввода графической информации*, что не позволяет в полной мере использовать ДВК, когда необходим ввод значительных объемов информации с графиков и эскизов. Наиболее удобным сред-

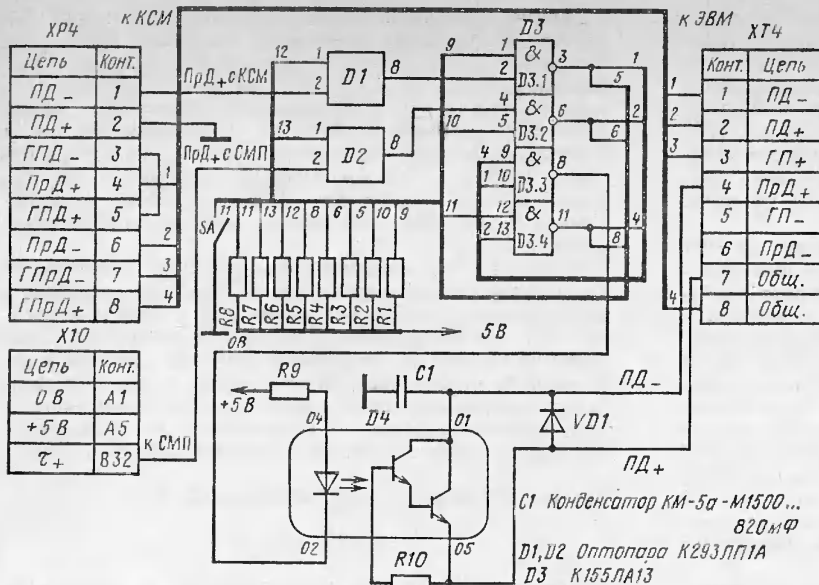
ством ввода такой информации являются планшетные устройства типа СМП6410, которые, несмотря на широкие возможности по передаче данных в различных режимах, тем не менее не могут быть непосредственно подключены к комплексу ДВК из-за отсутствия у них свободных портов ввода, а в старшей модели ДВК-3 и свободного гнезда для установки дополнительной платы с такими портами.

Предлагаемое техническое решение (рис. 1) не требует установки интер-

фейсного модуля. Информация в ДВК передается через существующий порт последовательного ввода консольного терминала. При этом режим работы СМП6410 устанавливается следующим: скорость передачи данных 9600 Бод/с; формат посылки 8 бит без бита паритета; стартовых битов — 1, стоповых — 2.

При посылке данных со стороны контроллера символьного монитора (КСМ) стартовый бит передается высоким уровнем, стоповые — низким. Для графического соотношения уровней стартового и стоповых битов обратное. Инвертор D3.2 приводит уровни сигналов с планшета к стандарту, принятому для КСМ. При замкнутом переключателе SA работа возможна только

* Тенденция развития диалоговых вычислительных комплексов / Кокорин В. С., Криданер Л. С., Попов А. А. и др. // Микропроцессорные средства и системы. — 1986. — № 4. — С. 11-15.



1. Контакты 14 D1, D2, D3 подсоединяются к А5
 2. Контакты 7 D1, D2, D3 подсоединяются к А1

- D1, D2 Оптопара К293ЛП1А
- D3 К155ЛА13
- D4 Оптопара А0Т110А
- R1...R8 МПТ-0,125-2,4 КОМ
- R9 МПТ-0,125-200 Ом
- R10 МПТ-0,125-100 КОМ
- VD1 Стабилитрон КС156А
- ХР4, ХТ4 розетка СНО53-8
- Х10 розетка СНО64-86Р

Рис. 1. Интерфейс между устройством ввода графики СМП6410 и ДВК3-М2

с клавиатуры (прохождение информации с СМП блокируется), при размыкнута — только с планшета.

Поскольку драйвер консольного терминала не обладает полной «прозрачностью», данные с графипаншета передаются программным способом. При обращении к метке INTD блокируются прерывания от консольного терминала, чем достигается передача каждого байта информации со стороны СМП, а информация с буферного регистра вводится по установке флага готовности в регистре состояния. Программа обслуживания приведена на рис. 2. Головная программа TSTSMP (рис. 3) testa устройства СМП6410 выполняет индикацию режима работы планшета, вывод координат X и Y, а также индикацию положения точки в графическом режиме. К обычному режиму работы терминал возвращается после завершения работы с графипаншетом

C MAIN TSTSMP.FOR

```

CALL INIT
CALL GRMODE(1)
CALL VIEWPO(0,0,399,279)
CALL WINDOW(0,0,3990,2790)
CALL DRAWMO(1)

CALL INTD

1 CALL INPL(NT,IX,IY)

CALL DOT(IX,IY)
TYPE 3,NT,IX,IY
3 FORMAT('+',IX,' NT=',I4,
1 ' X=',I5,' Y=',I5)
2 IF(NT.EQ.'100.AND.IX.EQ.8256
1 .AND.IY.EQ.8256) GOTO 4
GOTO 1

4 CALL INTE
STOP
EN

```

Рис. 3. Программа теста устройства СМП6410

```

.TITLE SMP
.GLOBLE INTD,INTE,INPL

TTRS == 177560
TTRB == 177562

; Запрет прерывания от IT
INTD: BIC #^B1000000, @#TTRS
RTS PC

; Разр. прерывания от IT
INTE: BIS #^B1000000, @#TTRS
RTS PC

; Адрес NT в R0
INPL: TST (R5)+
MOV (R5)+, R0

1x: TSTB @#TTRS
BPL 1x
MOV B @#TTRB, (R0)
CMPB #^B10000001, @#TTRB
BEQ 2x
CMPB #^B10000011, @#TTRB
BEQ 2x
CMPB #^B10000000, @#TTRB
BEQ 2x

; Векторн. режим ?
CMPB #^B10000010, @#TTRB
BEQ 2x
CMPB #100, @#TTRB
BEQ 2x
BR 1x

2x: MOV #2, R0

; Ввод коорд. X, Y
3x: TSTB @#TTRS
BPL 3x
MOV B @#TTRB, R4
4x: TSTB @#TTRS
BPL 4x
MOV B @#TTRB, R3
ASH #7, R3
ADD R3, R4
MOV R4, @ (R5)+
DEC R0
BEQ 5x
BR 3x

5x: RTS PC
.END

```

Рис. 2. Программа обслуживания графипаншета СМП6410

замыканием переключателя SA и вводом с клавиатуры символов «a» до появления сообщения, сигнализирующего о восстановлении режима работы по прерываниям.

Телефон 267-01-10, д. 5-74, Москва
 Статья поступила 12.02.88

УДК 681.327
 С. Н. Шустенко, В. С. Донеv

СПЕЦИАЛИЗИРОВАННАЯ МАГИСТРАЛЬ ДЛЯ АВТОНОМНЫХ ОКЕАНОЛОГИЧЕСКИХ ПРИБОРОВ

Для совместных исследований Институту океанологии им. П. П. Ширшова АН СССР и Институту океанологии Болгарской Академии наук потребовалось объединить усилия по созданию необходимой измерительной аппаратуры. Наиболее эффективна модульная организация приборов

на основе единой внутрисистемной магистрали. Структурно аппаратная часть автономных океанологических приборов в большинстве случаев состоит из управляющих, измерительных и вспомогательных функционально независимых модулей. Модули можно использовать в составе различных приборов.

Сигналы магистрали для управления работой модулей формируются управляющими модулями на микропроцессорах (МП) с минимальными аппаратными затратами.

На функциональный состав сигналов влияют специфические требования, предъявляемые к автономной океанологической аппаратуре: низкое энергопотребление, высокая надежность, малые габариты и масса. Указанные требова-

ния не позволяют использовать для построения управляющих модулей МП-структуры с типовой организацией. Поэтому перед созданием специализированной магистрали были разработаны взаимозаменяемые управляющие модули с пониженным энергопотреблением на основе различных МП (базовые — МК БИС серий КР580 (СССР) и СМ600 (НРБ), имеющие близкую структурную организацию, входные и выходные сигналы, что упрощает создание единой магистрали и на ее основе взаимозаменяемых модулей [1—3]).

Энергопотребление МП-модулей было снижено за счет максимального использования КМОП интегральных схем и введения программно-импульсного режима питания наиболее энергопотребляющих БИС [4]. Для этого режима в состав модуля вводится ключ, коммутирующий шину питания необходимых микросхем. Ключ выключается самим МП, после чего он оказывается вместе с другими микросхемами обесточенным. Питание включается либо таймером, либо модулями с постоянным питанием, работающими по заданной процессором программе. После включения МП определяет устройство, которое его включило, по выходным сигналам триггеров специального регистра состояния системы. Каждый триггер регистра конструктивно располагается в соответствующем модуле и устанавливается им при необходимости включения питания МП.

Внутрисистемная магистраль предназначена для создания измерительных приборов, работающих под управлением 8-битных МП. Открытая архитектура приборов позволяет подключать к магистрали различные измерительные и вспомогательные модули, в том числе и модули расширения базовой структуры управляющего модуля (дополнительную память, таймеры, устройства ввода-вывода информации, блоки сопряжения и т. п.). В состав магистрали входят сигналы для организации программно-импульсного режима питания модулей приборов. Число сигнальных линий должно быть оптимально для уменьшения габаритов и повышения надежности приборов (за счет сокращения числа межмодульных соединений) и для минимизации аппаратных затрат на адресацию и управление модулями и их узлами.

Перечисленные требования достаточно полно удовлетворяются 32 сигналами магистрали. Каждому сигналу из информационной, адресной и управляющей групп выделена отдельная линия (конструктивно магистраль выполнена в виде разъема на плате МП-модуля). Уровни напряжения логических сигналов совместимы с уровнями микросхем ТТЛ-серий: выходные токи низкого уровня — 1,8 мА, а высокого уровня — 0,2 мА, емкость нагрузки — 100 пФ. В ряде случаев к магистрали можно подключить непосредственно выходы БИС МП, выполненных по n-МДП-технологии.

Недостаток описываемой магистрали — отсутствие в ее составе сигналов требования и обработки прерываний из-за различных алгоритмов работы в этом режиме выбранных МП, что неоправданно увеличивает аппаратные затраты на приведение имеющихся сигналов к одному виду. В какой-то мере указанный недостаток можно компенсировать программно с помощью регистра состояния системы (его выходные сигналы — использовать для формирования приоритетных изменений алгоритма работы системы).

Телефон 124-85-13, Москва, Шустенко С. Н.

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. — М.: Радио и связь, 1984. — 272 с.
2. Станчев В. М., Бакърджиев Е. Т. Микропроцессорная система СМ600 — описание, программирование и приложение. — София: Техника, 1984. — 264 с.
3. Донов В. С. Микропроцессорный комплект БИС серии СМ600 // Микропроцессорные средства и системы. — 1988. — № 5. — С. 3—5.
4. Донов В. С. Импульсно захватываемое на микропроцессорной системе // Радио, телевизия, электроника. — 1987. — № 2. — С. 31—32.

Статья поступила 19.04.88

УДК 681.325.5

Н. Ф. Гринь, Т. Г. Уткина

МОДЕМ СВЯЗИ С БЫТОВЫМ МАГНИТОФОНОМ

Отлаживая большие программы, при отключении питания отладочных средств необходимо сохранять введенную информацию. Это существенно сокращает затраты времени на повторный ввод. Для простых автономных учебно-отладочных устройств [1] проблему легко решить, используя бытовой кассетный магнитофон для записи информации на магнитную ленту, хранения и последующего считывания.

Методы записи цифровой информации на магнитный носитель различны [2]. Наиболее часто используется метод двухчастотного кодирования [3, 4].

Предлагаемый способ записи на бытовой кассетный магнитофон в отличие от указанных представляет собой частотную модуляцию сигнала, при котором частота 3000 Гц соответствует Лог. 1, а отсутствие сигнала — Лог. 0. Аппаратно способ записи реализован с помощью D-триггера — фиксатора информации, электронного коммутатора и полосового фильтра (рис. 1).

Информация записывается в фикса-

тор по синхрипульсу, формируемому опорным генератором (3000 Гц), выход которого управляет электронным коммутатором. Таким образом, на вход магнитофона подается сигнал (с частотой опорного генератора), соответ-

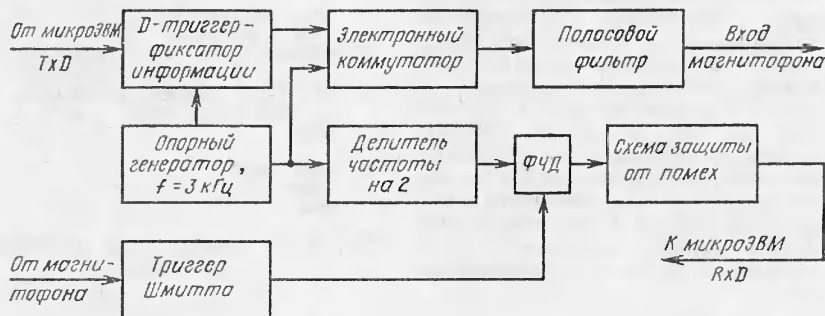


Рис. 1. Блок-схема модема

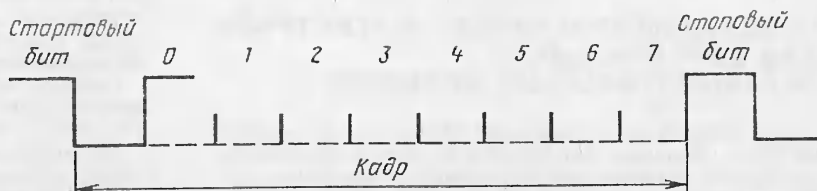


Рис. 2. Формат кадра асинхронной передачи данных

```

BA EQU 0 ; НАЧАЛЬНЫЙ АДРЕС
EAMB EQU 0FH ; СТАРШИЙ БАЙТ КОНЕЧНОГО АДРЕСА
EALB EQU 0FFH ; МЛАДШИЙ БАЙТ КОНЕЧНОГО АДРЕСА
SBD EQU 0CCH ; ВЕЛИЧИНА ДЛЯ ЗАДАНИЯ СКОРОСТИ
; ПРИЕМА/ПЕРЕДАЧИ 300 БОД

REGFL DSEG ;
DATA 2FH ;
BSEG ;

RI BIT SCON.0 ;
TI BIT SCON.1 ;
TR0 BIT SCON.4 ;
TR1 BIT SCON.6 ;

; CSEG
; *****
; ПОДПРОГРАММА ЗАПИСИ ИНФОРМАЦИИ НА МАГНИТОФОН
; *****
TXDCS: MOV R6,#EAMB ; ХРАНЕНИЕ СТАРШЕГО БАЙТА
; *****
MOV R5,#EALB ; ХРАНЕНИЕ МЛАДШЕГО БАЙТА
; *****
ACALL SPT1IN ; ИНИЦИАЛИЗАЦИЯ
; *****
; ПОСЛЕДОВАТЕЛЬНОГО КАНАЛА,
; ТАЙМЕРОВ
MOV DPTR,#BA ; УСТАНОВКА НАЧАЛЬНОГО АДРЕСА
MOVX A,@DPTR ; СЧИТЫВАНИЕ БАЙТА ИЗ ОЗУ
SETB TR1 ; ЗАПУСК ТАЙМЕРА 1

; ЦИКЛ ПЕРЕДАЧИ ИНФОРМАЦИИ ИЗ ОЗУ
REPR1: ACALL SPOUTB ; ПЕРЕДАЧА БАЙТА ИНФОРМАЦИИ
ADD A,R7 ; ПОЛУЧЕНИЕ КОНТРОЛЬНОЙ СУММЫ
MOV R7,A ; ХРАНЕНИЕ КОНТРОЛЬНОЙ СУММЫ

; СРАВНЕНИЕ ТЕКУЩЕГО АДРЕСА С КОНЕЧНЫМ
MOV A,R6 ; *****
CJNE A,DPH,NEXT ; ПЕРЕХОД ПРИ НЕРАВЕНСТВЕ
; *****
MOV A,R5 ; *****
CJNE A,DPL,NEXT ; ПЕРЕХОД ПРИ НЕРАВЕНСТВЕ
; *****
MOV A,R7 ; *****
ACALL SPOUTB ; ПЕРЕДАЧА КОНТРОЛЬНОЙ СУММЫ
; *****
RET

NEXT: INC DPTR ; ИНКРЕМЕНТ АДРЕСА ОЗУ
MOVX A,@DPTR ; СЧИТЫВАНИЕ БАЙТА ИЗ ОЗУ
SJMP REPR1 ; ПОВТОРЕНИЕ ЦИКЛА ПЕРЕДАЧИ

; *****
; ПОДПРОГРАММА СЧИТЫВАНИЯ ИНФОРМАЦИИ С МАГНИТОФОНА
; *****
RXDCS: ACALL SPT2IN ; ИНИЦИАЛИЗАЦИЯ
; *****
; ПОСЛЕДОВАТЕЛЬНОГО КАНАЛА,
; ТАЙМЕРОВ
MOV DPTR,#BA ; УСТАНОВКА НАЧАЛЬНОГО АДРЕСА
SETB TR1 ; ЗАПУСК ТАЙМЕРА 1
REPR1: JNB RI,REPR1 ; ПЕРЕХОД НА СЧИТЫВАНИЕ ФЛАГА
; ПРИЕМНИКА

CLR RI ; *****
MOV A,SBUF ; СЧИТЫВАНИЕ РЕГИСТРА ПРИЕМНИКА
MOVX @DPTR,A ; ЗАПИСЬ БАЙТА В ОЗУ
ADD A,R7 ; ПОЛУЧЕНИЕ КОНТРОЛЬНОЙ СУММЫ
MOV R7,A ; ХРАНЕНИЕ КОНТРОЛЬНОЙ СУММЫ
SETB TR0 ; ЗАПУСК ТАЙМЕРА 0

; ЦИКЛ ПРИЕМА ИНФОРМАЦИИ В ОЗУ
NEXTBT: JNB RI,SWAIT ; ПЕРЕХОД НА АНАЛИЗ ЗАДЕРЖКИ
CLR RI ; *****
MOV A,SBUF ; СЧИТЫВАНИЕ РЕГИСТРА ПРИЕМНИКА
INC DPTR ; ИНКРЕМЕНТ АДРЕСА ОЗУ
MOVX @DPTR,A ; ЗАПИСЬ БАЙТА В ОЗУ
ADD A,R7 ; ПОЛУЧЕНИЕ КОНТРОЛЬНОЙ СУММЫ
MOV R7,A ; ХРАНЕНИЕ КОНТРОЛЬНОЙ СУММЫ
CLR A ; *****
MOV TLO,A ; СБРОС СТАРШЕГО БАЙТА ТАЙМЕРА 0
MOV TH0,A ; СБРОС МЛАДШЕГО БАЙТА ТАЙМЕРА 0
SJMP NEXTBT ; ПОВТОРЕНИЕ ЦИКЛА ПРИЕМА

SWAIT: JNB TF0,NEXTBT ; ПЕРЕХОД, ЕСЛИ НЕТ
; ПЕРЕПОЛНЕНИЯ ТАЙМЕРА 0

; СРАВНЕНИЕ ПОДСЧИТАННОЙ КОНТРОЛЬНОЙ СУММЫ С ПЕРЕДАННОЙ
MOVX A,@DPTR ; *****
CLR A ; *****
SUBB A,R7 ; *****
XRL A,R7 ; СРАВНЕНИЕ КОНТРОЛЬНЫХ СУММ
JZ ADEC ; ПЕРЕХОД ПРИ РАВЕНСТВЕ

ADEC: ACALL ADDRDC ; ДЕКРЕМЕНТ АДРЕСА
; *****
; ПОДПРОГРАММА ПЕРЕДАЧИ БАЙТА
; *****
SPOUTB: JNB TI,SPOUTB ; ПЕРЕХОД НА СЧИТЫВАНИЕ ФЛАГА
; ПЕРЕДАТЧИКА

CLR TI ; *****
MOV SBUF,A ; ЗАГРУЗКА РЕГИСТРА ПЕРЕДАТЧИКА
RET

; *****
; ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ ПОСЛЕДОВАТЕЛЬНОГО КАНАЛА,
; ТАЙМЕРОВ
SPT2IN: CLR A ; *****
MOV TH0,A ; СБРОС СТАРШЕГО БАЙТА ТАЙМЕРА 0
MOV TLO,A ; СБРОС МЛАДШЕГО БАЙТА ТАЙМЕРА 0
MOV R7,#0 ; СБРОС РЕГИСТРА КОНТРОЛЬНОЙ
; СУММЫ
ORL P3,#1 ; ПРОГРАММИРОВАНИЕ КАНАЛА НА
; ВВОД
MOV SCON,#52H ; ИНИЦИАЛИЗАЦИЯ
; *****
MOV TMOD,#21H ; ПОСЛЕДОВАТЕЛЬНОГО КАНАЛА
MOV TH1,#SBD ; ИНИЦИАЛИЗАЦИЯ ТАЙМЕРОВ
; *****
RET ; УСТАНОВКА СКОРОСТИ
; ПРИЕМА/ПЕРЕДАЧИ

; *****
; ПОДПРОГРАММА ДЕКРЕМЕНТА АДРЕСА В DPTR
; *****
ADDRDC: MOV A,DPL ; *****
JNZ DLOWA ; *****
DEC DPH ; *****
DEC DPL ; *****
RET

DLOWA: MOV A,DPL ; *****
JNZ DLOWA ; *****
DEC DPH ; *****
DEC DPL ; *****
RET

```

Рис. 3. Исходные тексты подпрограмм записи и считывания

вующий появлению информации с уровнем Лог. 1 на входе модема. При передаче Лог. 0 сигнал на входе отсутствует.

Обмен информацией проходит в асинхронном последовательном режиме с переменной скоростью передачи и 10-битным кадром, состоящим из стартового бита нулевого значения, восьми бит данных (передача и прием которых начинается с младшего бита) и стопового бита единичного значения (рис. 2).

Информация записывается на магнитную ленту блоками произвольной длины. Каждому блоку в течение 2 с должен предшествовать тон маркера частотой 3000 Гц. В конце блока информации записывается контрольная сумма.

Исходные тексты подпрограмм записи и считывания информации приводятся ниже. Подпрограмма TXDCS, включающая в себя подпрограммы SPT1IN и SPOUTB, считывает информацию из ОЗУ отладочного средства, записывает ее в буферный регистр передатчика последовательного канала связи и подсчитывает контрольную сумму передаваемой информации (рис. 3).

Подпрограмма SPT1IN программиру-

ет последовательный канал и задает режим работы таймера 1, изменяя этим скорость передачи. В тексте подпрограммы указана величина для задания скорости передачи 300 Бод при тактовой частоте 6 МГц.

Подпрограмма SPOUTB организует передачу байта информации в буферный регистр передатчика.

Подпрограмма RXDCS записывает информацию в ОЗУ отладочного средства и контролирует принимаемую информацию. При несовпадении подсчитанной контрольной суммы с переданной устанавливаются флажки, сигнализирующие об ошибочном считывании. Выход из подпрограммы RXDCS происходит при обнаружении тона маркера частотой 3000 Гц в течение 0,1 с.

Подпрограмма ADDRDC включает в себя подпрограммы SPT2IN и ADDRDC. Подпрограмма SPT2IN задает режим работы последовательному каналу, готовит таймер 0 для организации временной задержки и задает режимы работы таймера 0 и 1.

Подпрограмма ADDRDC декрементирует адрес ОЗУ отладочного устройства.

Перед считыванием информации с магнитофона необходимо по счетчику ленты установить нужный участок лен-

ты. Модем подключается к линейному выходу магнитофона стандартным кабелем.

При считывании данных с магнитофона (рис. 1) сигнал с его выхода поступает на вход триггера Шмитта, преобразующий входной сигнал в цифровой вид. Далее преобразованный сигнал проходит на вход фазочастотного дискриминатора (ФЧД), где сравниваются частоты входного сигнала и опорного генератора. При совпадении указанных частот воспроизводимая информация поступает на вход схемы защиты от помех. Несовпадение фаз устраняется триггером, входящим в состав ФЧД.

Схема защиты от помех — это регистр сдвига со сравнением; служит для защиты модема от сбоев при появлении ложных одиночных положительных импульсов (например, при дефектном покрытии ленты) или при потере импульса (например, в случае склейки ленты).

При появлении помехи на входе схемы защиты информация на ее выходе останется неизменной, т. е. считывание информации будет правильное. Элементная база модема (рис. 4) — интегральные микросхемы серий К561, К140, КР1006.

Достоинства модема — отсут-

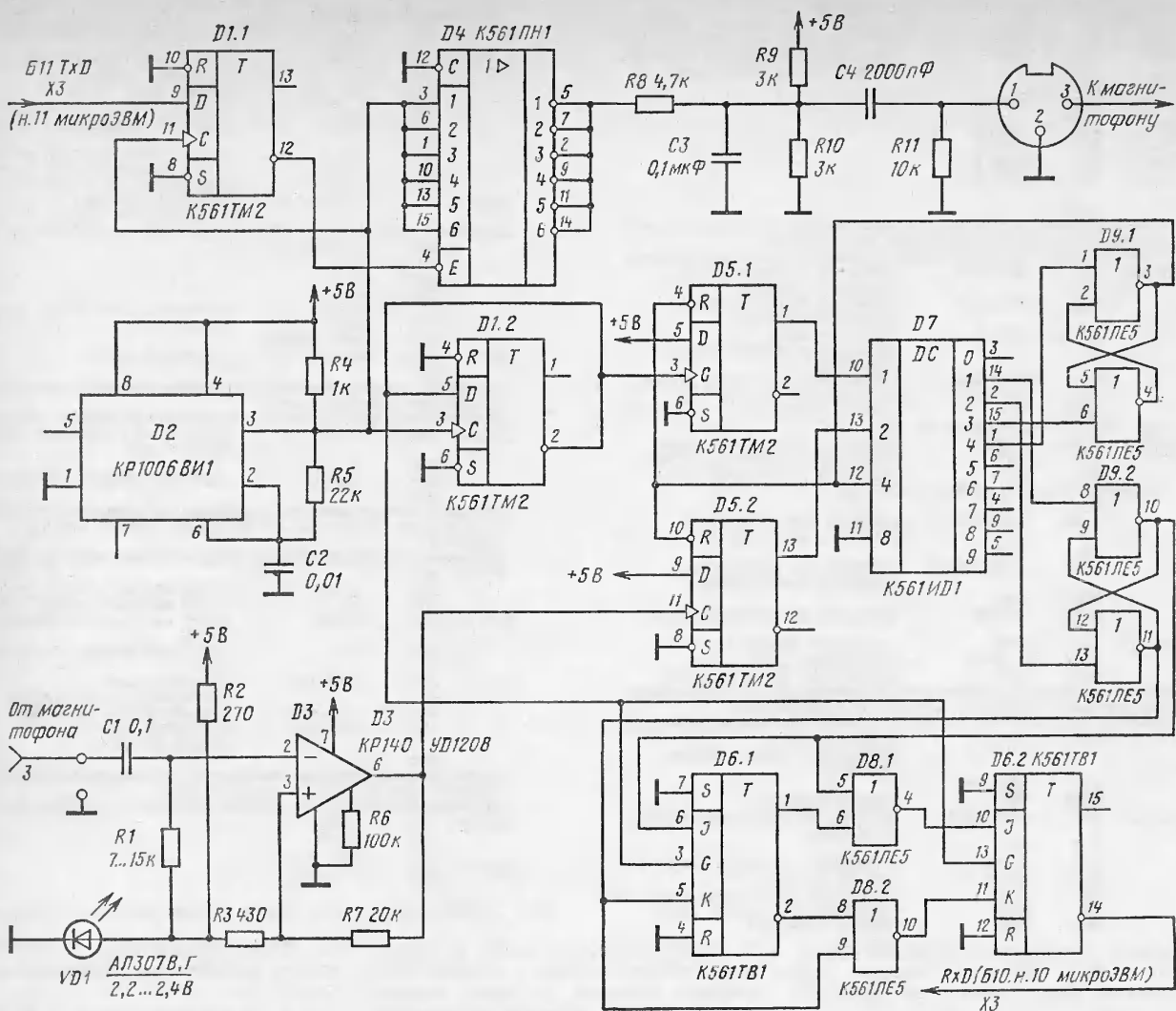


Рис. 4. Принципиальная схема модема

ЛИТЕРАТУРА

1. Электронная промышленность.— 1986.— № 9.— С. 8—10.
2. Каган Б. М. Электронные вычислительные машины и системы.— М.: Энергия, 1979.— С. 156—160.
3. Алексенко А. Г., Галицын А. А., Иванников А. Д.

Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.— С. 182—186.

4. Микропроцессорные средства и системы.— 1986.— № 3.— С. 60—61.

Телефон 468-81-75, Москва

Статья поступила 26.01.88

РЕКЛАМА

Государственная публичная научно-техническая библиотека СССР подготовила к печати «УКАЗАТЕЛЬ ЗАРУБЕЖНЫХ ОРГАНИЗАЦИЙ, ИЗДАЮЩИХ ЛИТЕРАТУРУ ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ».

УКАЗАТЕЛЬ подготовлен по материалам, опубликованным в мировой печати в 1985—1989 гг.

В УКАЗАТЕЛЬ включены сведения о 1500 зарубежных организациях: издательских и промышленных фирмах, научных и исследовательских институтах, центрах и обществах; факультетах, кафедрах, лабораториях, вычислительных центрах уни-

верситетов и колледжей, издающих литературу по вычислительной технике или осуществляющих разработки в области вычислительной техники и публикующих результаты исследований в собственных изданиях; даны их адреса.

УКАЗАТЕЛЬ предназначен для работников библиотек и органов информации, занимающихся комплектованием иностранной литературы и международным книгообменом, подготавливающих обзорную информацию, а также для организаций, ориентирующихся на международный рынок.

Издание снабжено алфавитным указателем периодических и продолжающихся

изданий.

УКАЗАТЕЛЬ выйдет из печати в IV кв. 1989 г. Подписная стоимость 12 руб.

Заявку на УКАЗАТЕЛЬ надо направлять в ГПНТБ СССР. Просим в заявке указать № _____ перечисления от « » _____ 89 г. и перечисленную сумму.

Расчетный счет ГПНТБ СССР № 052110732 в Дзержинском отделении Жилсоцбанка г. Москвы. Деньги за указатель можно отправить почтовым переводом на указанный счет.

Для иногородних подписчиков: Дзержинское отделение Жилсоцбанка г. Москвы — код 74 МФО 201638.

**ДОРАБОТКИ БИС
СЕРИИ КР588**

При разработке микропроцессорных систем на базе БИС серии КР588 выпуска 1985—1987 гг. приходится сталкиваться с несоответствием технических характеристик микросхем требованиям технической документации [1]. Предприятие-изготовитель БИС серии КР588 не ставит в известность предприятия-потребители о недоработках в поставляемых БИС. В публикациях, описывающих работу изделий, выполненных на базе БИС серии КР588, об изменениях или ограничениях в работе БИС этой серии не сообщалось [2—5].

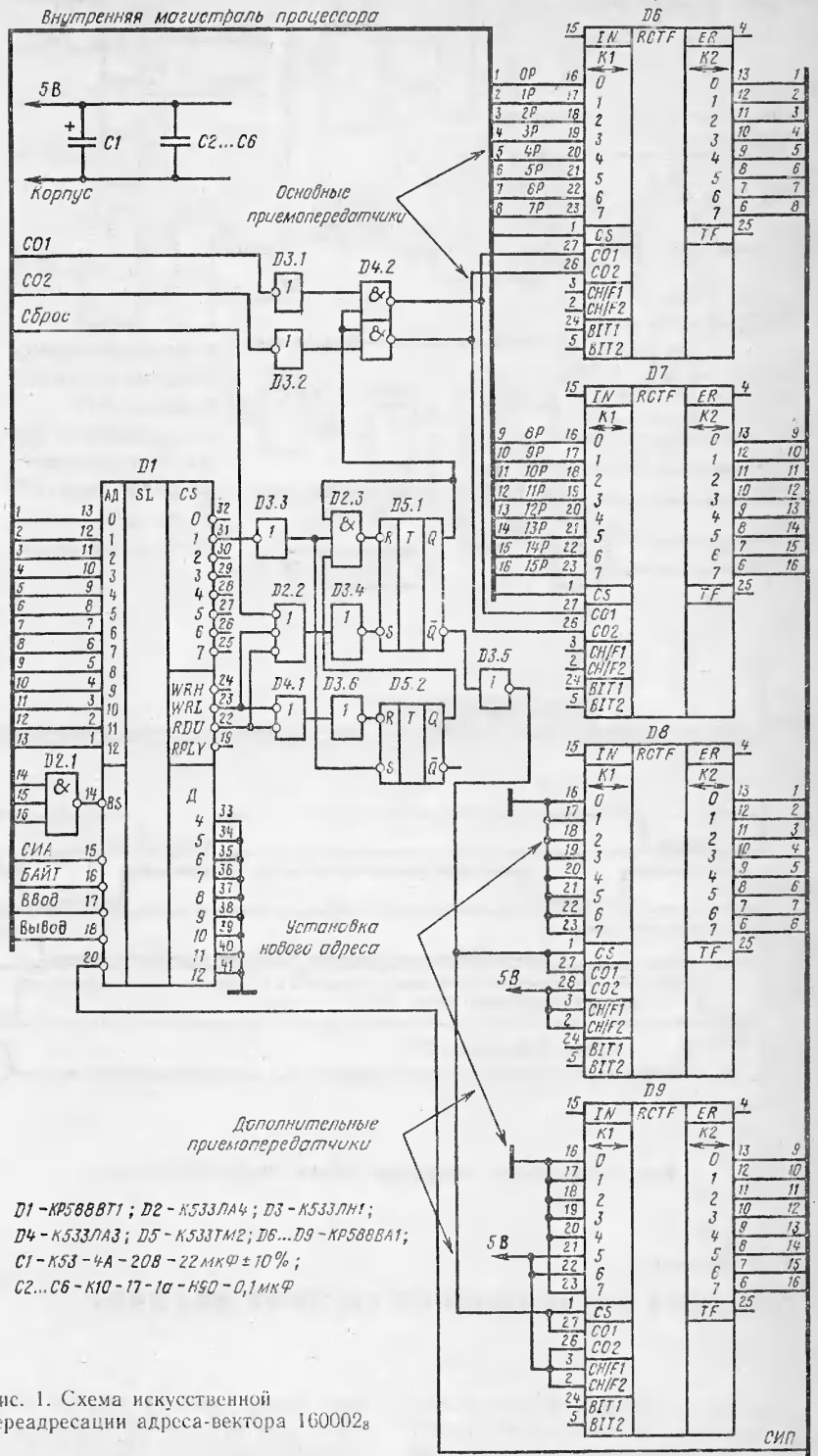
Из выявленных недоработок, рассмотрим три, обнаруженные при отладке МП, собранного на БИС серии КР588, и схемные решения для устранения двух из них:

1. Линия «Останов» (HALT) МП (БИС КР588ВГ1 — контакт 36) обслуживается адресом-вектором 16000₈, а не 6000₈, как указано в технической документации [1]. Задействовать эту линию с адресом 6000₈ без доработки схемы МП не представляется возможным, так как адрес 16000₈ находится в области адресов ВУ.

Необходима искусственная переадресация адреса-вектора 16000₈ с помощью селектора адреса (SL — D1), формирующего сигнал по коду 16000₈, двух дополнительных приемопередатчиков (RCTF—D8, D9) с жесткой установкой преобразованного адреса (6000₈) и элементов управления (D2...D5) основными (RCTF—D6, D7) и дополнительными приемопередатчиками (рис. 1). При наличии кода адреса 16000₈ основные приемопередатчики закрыты, а дополнительные открыты для передачи нового адреса-вектора.

2. Если при работе МП в режиме внешних прерываний с фиксированными адресами-векторами или прямого доступа к памяти сигналы требования прерывания (INT0...INT3) и требования прямого доступа (DMR) совпадают по времени с выполнением команд расширенной арифметики (DIV, ASH, ASHC), то МП прекращает работу. Этот дефект можно устранить путем блокировки поступления в МП сигналов INT0...INT3 и DMR на время выполнения указанных команд. Схема блокировки приведена на рис. 2, временная диаграмма — на рис. 3.

Работа схемы заключается в формировании сигнала запрета на поступление в МП сигналов INT0...INT3 и DMR во время выполнения команд DIV, ASH, ASHC. Сигнал запрета образован логической суммой вырабатываемых МП сигналов KB1 (БИС КР588ВС2 — контакт 32), ОРА1 (БИС КР588ВУ2-0001, БИС КР588ВУ2-0002, БИС КР588ВУ2-0006 — контакт 41),



- D1 - КР588ВГ1; D2 - К533ЛА4; D3 - К533ЛН1;
- D4 - К533ЛА3; D5 - К533ТМ2; D6...D9 - КР588ВВ1;
- C1 - К53-4А-20В - 22 мкФ ± 10%;
- C2...C6 - К10-17-1а - Н90 - 0,1 мкФ

Рис. 1. Схема искусственной переадресации адреса-вектора 16000₈

ОРА2 (БИС КР588ВУ2-0004, БИС КР588ВУ2-0007 — контакт 41), ENDNG (БИС КР588ВУ2 — контакт 38) и сформированным из кодов (KDA10...KDA15) команд (DIV, ASH, ASHC) сигналом с дешифратора (D4, D5).

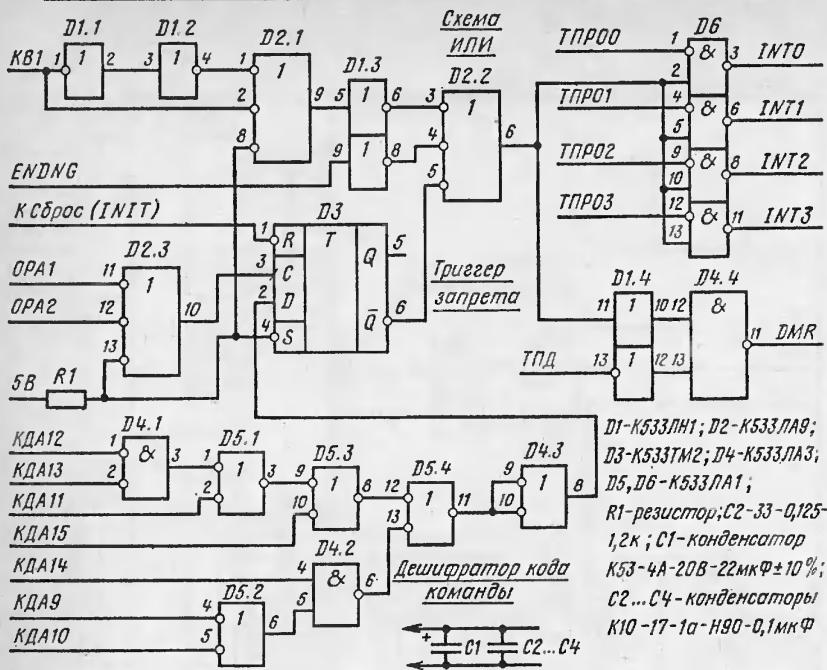


Рис. 2. Схема блокировки прохождения сигналов INTO...INT3 и DMR

3. Вместо двух режимов начального пуска в МП реализован только один: с выборкой счетчика команд из ячейки 24₈ и слова состояния (ССП) МП из ячейки 26₈ [1]. Режим с выборкой счетчика команд из ячейки 17300₈ и установкой Р-бита ССП в 1 и Т-бита ССП в 0 не реализован и организовать его искусственно не представляется возможным [1].

Рассмотренные недостатки подтверждены ведущими специалистами предприятия-изготовителя.

252141, Киев, ул. Соломенская, 41, кор. 2, кв. 96, Сунгурову В. С.

ЛИТЕРАТУРА

1. Микросхема К588ВУ2, Н588ВУ2. Техническое описание. ЭРЗ.482.033—01 ТО1.— С. 18.
2. Черняковский Д. Н., Шиллер В. А., Юровский А. А. Процессор с системой команд и интерфейсом микроЭВМ «Электроника 60» на основе БИС серии КР588 // Электронная промышленность.— 1983.— № 9.— С. 11—13.
3. Громов Г. Н., Дроздов Р. В., Криворучко Ю. Т., Огородников В. Н. Вычислитель на микропроцессорных БИС серии К588 // Электронная промышленность.— 1983.— С. 22—23.
4. Корнупов Л. П., Криворучко Ю. И., Пеляк Е. Л. Одноплатая микроЭВМ на БИС серии К588 // Электронная промышленность.— 1986.— № 2.
5. Бобков В. А., Чернуха Б. Н., Свиридович В. С., Ключков В. П. Расширенный микропроцессорный комплект БИС серии К588 // Микропроцессорные средства и системы.— 1987.— № 1.— С. 6—7.

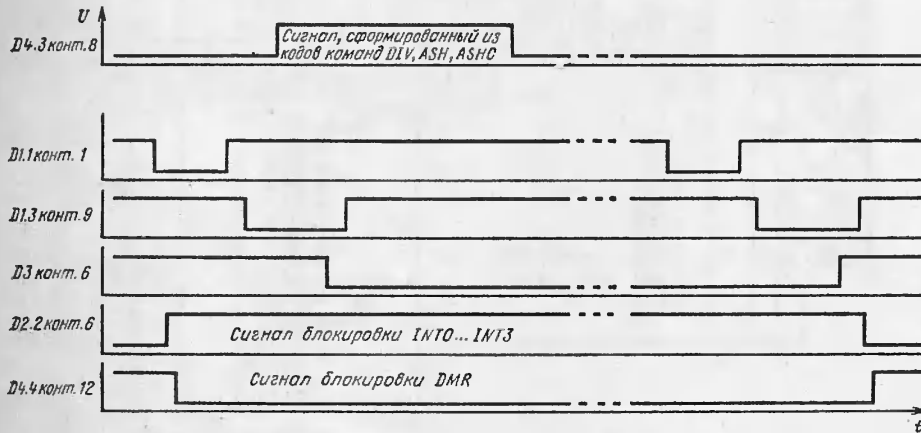


Рис. 3. Временная диаграмма работы схемы блокировки

УДК 681.3.06

У. Ф. Фейзханов

ОШИБКА В ОПЕРАЦИОННОЙ СИСТЕМЕ ДЛЯ ДВКЗ

Во многих случаях бывает полезно организовать взаимодействие оператора с вычислительным процессом так, чтобы он мог управлять программой с помощью клавиатуры терминала, но чтобы процесс не приостанавливался во время общения (например, эмулирующие приборы, игровые и вычислительные программы, выдающие теку-

щую информацию по запросу оператора). Для реализации такого взаимодействия средствами ОС RT-11 в слове состояния программы JSW, расположенном по адресу 44, устанавливается значение 50100. Затем по мере выполнения программы необходимо время от времени делать запросы к буферу клавиатуры I-ITINR (). При

этом переменная I принимает значение кода символа, введенного с клавиатуры, или значение меньше нуля, если символ не введен, т. е. программа не приостанавливается до ввода символа с клавиатуры. Пример такой программы, написанной на Фортране, приведен на рис. 1.

К сожалению, при работе с этой программой на ДВКЗ в однозначной ОС RT-11SJ появляется сообщение:

?Err 61 Illegal memory reference in routine "TEST" line 5 (PC=155136)

Это связано с особенностью реализации арифметических команд плавающей запятой (FIS) в микропроцессоре K1801BM2 и процедурой возврата из прерываний в ОС. Команды FIS выполняются программно. Эти программы расположены в «теневом» ПЗУ, где кроме этого имеются тесты, загрузчики и программа работы с пультовым терминалом. ПЗУ находится в адресах внешних устройств, но программе пользователя в явном виде адреса не доступны (поэтому ПЗУ называется «теневым»). Когда происходит прерывание во время выполнения команды-программы FIS, в стек записывается слово состояния процессора (при этом устанавливается восьмой бит, который указывает, что прерывание произошло из «теневого» ПЗУ).

```
PROGRAM TEST
CALL IPOKE ("44,"50100)
A=0.
DA=0.01
A=A+DA
I=ITINR ( )
IF (I.LT.0) GOTO 1
TYPE *,A
GOTO 1
END
```

Рис. 1.

Подпрограмма обработки прерываний от терминала и программы обработки командных прерываний EMT расположены в резидентном мониторе ОС (этим терминал отличается от других устройств ввода-вывода, подпрограммы обслуживания которых находятся в соответствующих драйверах). Для экономии места в оперативной памяти программы обработки командных прерываний и прерываний от терминала используют подпрограммы сохранения и восстановления регистров общего назначения SVREG и POPREG.

```
SAVEST=12
OLDPS=16
SVREG:  MOV    R2,-(SP)
        MOV    R3,-(SP)
        MOV    R4,-(SP)
        MOV    R5,-(SP)
        MOV    R0,-(SP)
        JMP    @R1
POPREG: MOV    SAVEST+4(SP),R3
        CMPB   #374,-2(R3)
        BLOS  NOPOP
        ASL    R2
        BEQ    NOPOP
        MOV    SP,R3
        ADD    #OLDPS+4,R3
        ADD    R3,R2
1R:     MOV    -(R3),-(R2)
        CMP    R3,SP
        BHI   1R
        MOV    R2,SP
        TST   (SP)+
        MOV    (SP)+,R0
        MOV    (SP)+,R5
        MOV    (SP)+,R4
        MOV    (SP)+,R3
        MOV    (SP)+,R2
        RTS   R1
```

Рис. 2.

Подпрограмма SVREG (рис. 2) заносит содержимое регистров в стек с обращением JSR R1, SVREG. После выполнения этой программы данные в стеке организуются, как показано на рис. 3. Затем следует сама программа обработки прерываний, которая завершается обращением к подпрограмме POPREG (см. рис. 2), восстанавли-

после SVREG SP->

перед POPREG SP->

Рис. 3.

вающей значения регистров из стека. Кроме того, эта подпрограмма переходит от системного стека (расположенного в адресах ниже ОС) к программному (расположенному обычно с адреса 1000) и наоборот. Условие перехода: код командного прерывания EMT больше или равен 374 и регистр R2 не равен нулю.

Когда прерывание от клавиатуры происходит во время операции FIS, при попытке определить, с каким кодом произошло командное прерывание EMT, возникает прерывание по неверному обращению к памяти. Действительно, как уже говорилось выше, программы FIS находятся в «теневом» ПЗУ (адресах внешних устройств) и в явном виде не доступны. Для других процессоров, где команды FIS реализованы на микропрограммном уровне, например для процессора M2 микроЭВМ «Электроника 60», такая ситуация не наблюдается. При прерывании во время выполнения команды FIS в стек заносится адрес текущей команды FIS, а не следующей, как при «нормальном» прерывании, и при возврате команда FIS начинается заново, а не продолжается, как в микропроцессоре K1801BM2. Некоторая некорректность, связанная с тем, что при прерывании от клавиатуры программа POPREG пытается определить код командного прерывания, устраняется, так как регистр R2 всегда равен нулю.

Предлагается несколько способов, предотвращающих такую ошибочную ситуацию:

1. Не использовать команды FIS, а реализовать их программно. Если есть возможность, то провести генерацию трансляторов с кодом FIS.

2. Вместо однозадачного монитора SJ применить двухзадачный FB. В RT-11FB резидентный монитор реализован несколько по-другому, и поэтому такие эффекты не возникают.

3. Внести исправления в ОС SJ: переставить некоторые строки в программе POPREG так, чтобы сначала проверить значение R2, а затем, если необходимо, определить код EMT. Измененная часть показана на рис. 4. Если нельзя провести генерацию ОС системы, то можно модифицировать уже

!СЛОВО СОСТОЯНИЯ ПРОЦЕССОРА
!АДРЕС ВОЗВРАТА ИЗ ПРЕРЫВАНИЯ
!R1
!R2
!R3
!R4
!R5
!R0
!

```
POPREG: ASL    R2
        BEQ    NOPOP
        MOV    SAVEST+4(SP),R3
        CMPB   #374,-2(R3)
        BLOS  NOPOP
```

Рис. 4.

```
RUN SIPP<CR>
*SY:RT11FB.SYS<CR>
Base? 0<CR>
Offset? 0<CR>

Base      Offset      Old      New?
000000    000000    000000    ;<CR>
Search for? 16603<CR>
Start?    0<CR>
End?      <CR>
Found at 042720
Found at 047574

Base      Offset      Old      New?
000000    000000    000000    ^Z<CR>
Offset? 47574<CR>

Base      Offset      Old      New?
000000    047574    016603    006302<CR>
000000    047576    000016    001416<CR>
000000    047600    122763    016603<CR>
000000    047602    000374    000016<CR>
000000    047604    127776    127763<CR>
000000    047606    101412    000374<CR>
000000    047610    006302    177776<CR>
000000    047612    001410    101410<CR>
000000    047614    010603    ^Y<CR>
*^<CR>
.BOD SY:
```

Рис. 5.

существующую. Для этого используются программы, корректирующие двойные файлы, например SIPP.SAV. Протокол модификации представлен на рис. 5. Конкретные адреса, где располагается программа POPREG, могут быть другими.

Телефон 535-08-06, п. Менделеево
Моск. обл.

Статья поступила 25.05.88

Назначение выводов микросхемы КР1801РЕ2

Вывод	Обозначение	Тип	Назначение
1	RD	Вход	Чтение
2	AN	«	Ответ
3	SYN	«	Синхронизация
4...11	AD4...AD11	Входы-выходы	Разряды адреса данных
12	GND	—	Общий
13...16	AD3...AD0	Входы-выходы	Разряды адреса данных
17...20	AD12...AD15	«	«
21, 22	—	—	Не используются
23	CS	Вход	Выбор микросхемы
24	U _{CC}	—	Напряжение источника питания

**ПОСТОЯННОЕ
ЗАПОМИНАЮЩЕЕ
УСТРОЙСТВО КР1801РЕ2**

Микросхема КР1801РЕ2 представляет собой ПЗУ емкостью 65536 бит с организацией 4096×16 разрядов, выполненное по полупроводниковой технологии на p-канальных МОП-транзисторах. Изготавливается в 24-выводном корпусе типа 239.24-1. Входит в состав микропроцессорного комплекта серии К1801 наряду с микропроцессорами К1801ВМ1, КМ1801ВМ2, КМ1801ВМ3, интерфейсными микросхемами К1801ВП1-030, К1801ВП1-033, К1801ВП1-034, К1801ВП1-035 и РПЗУ с УФ-стиранием К573РФ3.

Условное графическое обозначение микросхемы дано на рис. 1, назначение выводов показано в табл. 1, структурная схема приведена на рис. 2, временная диаграмма работы — на рис. 3. Входы и выходы в микросхеме совмещены, поэтому данные передаются в мультиплексном режиме.

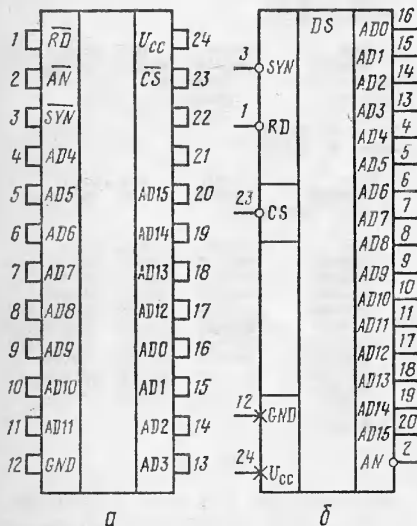


Рис. 1. Условное графическое обозначение микросхемы КР1801РЕ2 по порядку расположения (а) и функциональному назначению (б) выводов

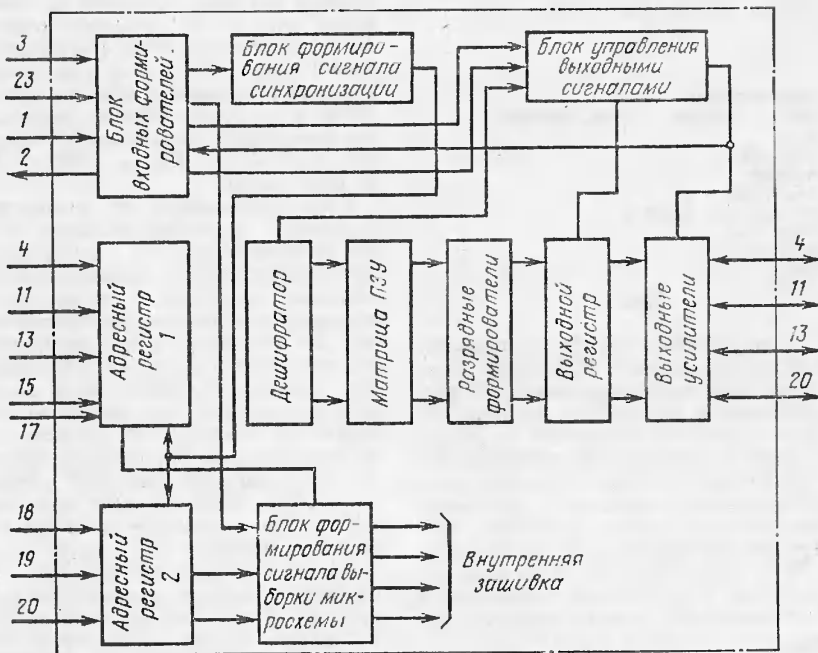


Рис. 2. Структурная электрическая схема КР1801РЕ2

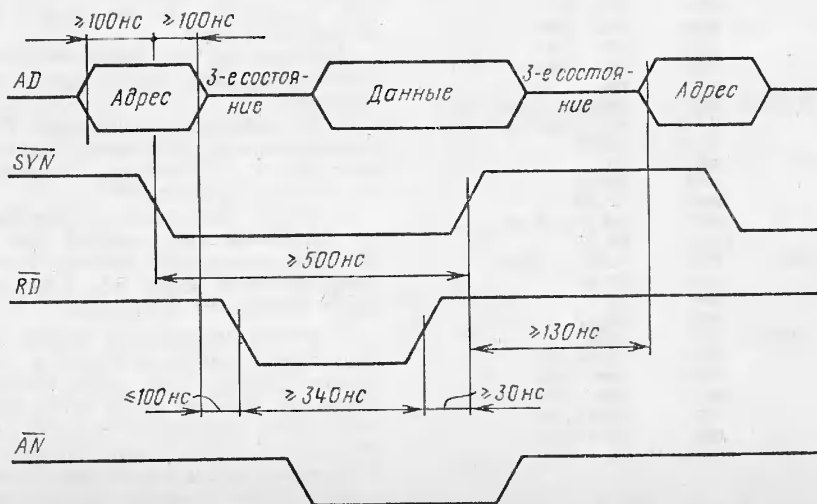


Рис. 3. Временная диаграмма работы КР1801РЕ2

Электрические характеристики микросхемы КР1801РЕ2

Параметр, единица измерения	Обозначение	Норма [макс. (мин.)]	
		КР1801РЕ2А	КР1801РЕ2Б
Выходное напряжение, В низкого уровня* высокого уровня**	U_{OL}	0,5	0,5
	U_{OH}	(24)	(24)
Потребляемая мощность, мВт, в режиме хранения обмена	P_{CCS}	220	220
	P_{CCD}	330	330
Входной ток, мкА высокого уровня низкого уровня	I_{IH}	15	15
	I_{IL}	15	15
Время выборки разрешения, нс	t_{CO}	300	500

* При $I_0=3,2$ мА.
** При $I_0=400$ мкА.

Таблица 2

Для управления работой используются четыре сигнала:

SYN — «синхронизация», обеспечивает запись адреса во входной регистр;
RD — «разрешение чтения», осуществляет выдачу считанной информации на общую магистраль при наличии сигнала «синхронизация»;

AN — «ответ», сопровождает информацию, поступающую из ПЗУ на общую магистраль;

CS — «выбор кристалла», гарантирует дополнительную выборку.

Наличие 3-разрядного программируемого интерфейса (адресные входы А13...А15) позволяет включать параллельно до восьми микросхем ПЗУ. Для выборки слова из ПЗУ необходимо подать код адреса слова на входы А13...А15. Вход АD0 не участвует в выборке адресов. Данные снимаются с выходов АD0...АD15. Основные электрические характеристики микросхемы КР1801РЕ2 приведены в табл. 2.

Г. Глушкова

РЕФЕРАТЫ СТАТЕЙ

УДК 681.325.5—181.4

Кобылинский А. В., Калатинец В. М., Заика А. И. **Микропроцессор КМ1810ВМ87 с плавающей запятой** // Микропроцессорные средства и системы.— 1989.— № 5.— С. 2.

Рассмотрены устройства, архитектура, типы данных и принцип работы микропроцессора с плавающей запятой.

УДК 681.322—192

Антонюк Б. В., Омельчук И. В., Присяжнюк В. Н., Терешенко Г. Г. **Тестовый процессор для мультипроцессорных систем** // Микропроцессорные средства и системы.— 1989.— № 5.— С. 13.

Рассматривается процессор, обеспечивающий селективный контроль информации, передаваемой по распределенной магистрали. Приведены структурная схема, временная диаграмма и описание работы процессора, предназначенного для контроля функционирования мультипроцессорных систем.

УДК 681.3

Бурдин В. Р., Карасев В. И., Коломыц В. Г., Макагон С. С., Чернявский А. Д. **Распределенная обработка данных с использованием СУБД КВАНТ** // Микропроцессорные средства и системы.— 1989.— № 5.— С. 22.

Описан программно-аппаратный комплекс, предназначенный для построения распределенной системы обработки данных, реализующей обращение к СУБД КВАНТ из программы на ДВКЗ, работающей в среде ОС РАФОС. Рассмотрены основные компоненты комплекса и их взаимодействие.

УДК 681.3.06

Корытный И. М., Мукалов И. О. **Объектно-ориентированная реализация диалога при доступе к реляционной базе данных** // Микропроцессорные средства и системы.— 1989.— № 5.— С. 28.

Характеризуется проектирование диалога реляционной СУБД, основанного на принципе непосредственного редактирования информации. Получены три представления объектов реляционной БД, адекватных операциям преобразования.

УДК 681.325.5—181.4

Kobylynsky A. V., Kalatinets V. M., Zaika A. I. **КМ1810ВМ87 floating-point arithmetic coprocessor** // Microprocessor devices and systems.— 1989.— N. 5 — P. 2.

The architecture, block diagram and operation principles of the FP microprocessor are explained. Numeric data formats processed by the device are shown.

УДК 681.322—192

Antonyuk B. V., Omelchuk I. V., Prisyazhnyukh V. N., Tereshenko G. G. **Test processor for multiprocessor systems** // Microprocessor devices and systems.— 1989.— N. 5.— P. 13.

The test unit providing selective data monitoring in the multiprocessor bus is described. The block diagram, timing and algorithm of the processor designed for monitoring multiprocessor system operation are explained.

УДК 681.3

Burdin V. R., Karasev V. I., Kolomyts V. G., Makagon S. S., Chernyavsky A. D. **Distributed data processing using KVANT database manager** // Microprocessor devices and systems.— 1989.— N. 5 — P. 22.

The software/hardware complex designed for distributed data processing is described. It provides access to KVANT database from the programs running on ДВК—3 — microcomputer under РАФОС operating system. The main components of the complex and their interaction are explained.

УДК 681.3.06

Korytny I. M., Mukalov I. O. **Direct Information editing as a dialogue access mechanism in a relational database system** // Microprocessor devices and systems.— 1989.— N. 5.— P. 28.

The design of the dialogue access in a relational database system based on direct visual information editing method is described. Three forms of data display (access, editing, view) are found to be most adequate for fundamental database management operations.

УДК 681.327.22

Данилов Н. Ю., Полукаров А. Д., Столяров М. В.
— Дисплейные микроконтроллеры семейства МИДИКОН // Микропроцессорные средства и системы.— 1989.— № 5.— С. 38.

Описаны микроконтроллеры, предназначенные для создания алфавитно-цифровых, графических и комбинированных цветных и монохромных дисплеев. Приведены технические характеристики, рассмотрены конструктивные и схемотехнические особенности, функциональные возможности, области применения микроконтроллеров.

УДК 681.327.23

Бучнев А. А., Сизых В. Г., Минин В. Ф.
Цветная полутонная станция профессионального назначения ГАММА-7.1 // Микропроцессорные средства и системы.— 1989.— № 5.— С. 45.

Приводится описание двухпроцессорной дисплейной станции ГАММА-7.1, выполненной в растровой технологии. Описаны функции дисплейного процессора, которым может быть любая микроЭВМ с магистралью МПИ. Представлена структура графического процессора, реализованного на микропроцессорном комплекте серии 1804, осуществляющего функционально-растровое преобразование графических примитивов. Описаны организация видеопамати и таблицы цветности. Приведено общее описание базового программного обеспечения дисплейного и графического процессоров.

УДК 681.3.06

Добриневский С. Ф. Процедуры обработки прерываний в Паскале ОСДВК // Микропроцессорные средства и системы.— 1989.— № 5.— С. 45.

Рассмотрены основные характеристики процедур обработки прерываний для системы программирования Паскаль ДВК, входящих в состав пакета INPROC. Описываются простейшие способы применения этих процедур, в том числе совмещение пользовательской и системной обработки прерываний.

УДК 681.327

Падирыков Ю. А., Белинский В. Т., Журило В. А. Модуль системы обработки данных реального времени // Микропроцессорные средства и системы.— 1989.— № 5.— С. 62.

Описывается устройство, выполненное на базе серийно выпускаемых процессоров «Электроника МТ-70» и «Электроника МС 1201.02», предназначенное для построения цифровых систем обработки данных в реальном масштабе времени.

УДК 681.325

Щелкунов Н. Н., Дианов А. П. Организация однокристальных микроконтроллеров // Микропроцессорные средства и системы.— 1989.— № 5.— С. 76.

Рассматриваются особенности однокристальных 8-разрядных микроконтроллеров серий iMCS-48, iMC-51 и K1816. Приводятся основные технические характеристики устройств, встроенных в МК.

UDC 681.327.22

Danilov N. Yu., Polukarov A. D., Stolyarov M. V. Display microcontroller family MIDICON. // Microprocessor devices and systems.— 1989.— N. 5 — P. 38.

MIDICON microcontrollers are designed to control alphanumeric, graphic and combined monochrome and colour displays. The technical features, constructive and circuit peculiarities, functions and possible application areas of microcontrollers are discussed.

UDC 681.327.23

Buchnev A. A., Sizykh V. G., Minin V. F. Professional colour semi-tone graphic workstation GAMMA-7.1. // Microprocessor devices and systems.— 1989.— N. 5.— P. 45.

The bit-map two-processor display workstation GAMMA-7.1 is described. The display processor functions are described. They may be performed by any Q-bus microcomputer. The block diagram of bit-slice graphic processor using K1804 (AMD29XX) microprocessor LSI family is shown. The graphic processor performs reinitializing of graphic primitives. The video RAM mapping and colour palettes organization are explained. The general description of software support for display and graphic processors is also included.

UDC 681.3.06

Dobrinevsky S. F. Interrupt processing procedures in DVK Pascal. // Microprocessor devices and systems.— 1989. N. 5 — P. 45.

The IPROC toolkit procedures for interrupt handling in the DVK Pascal are discussed. The simple applications of these procedures are explained, including combination of interrupt handling on user and system levels.

UDC 681.327

Padiryakov Yu. A., Belinsky V. T., Zhurilo V. A. A real-time data processing module. // Microprocessor devices and systems.— 1989.— N. 5 — P. 62.

The real-time data processor incorporating industrial "Elektronika MC 1201.02" microcomputer and "Elektronika MT-70" matrix processor is described. The unit is suitable for high-speed digital data processing.

UDC 681.325

Shelkunov N. N., Dianov A. P. Single-chip microcontrollers architecture. // Microprocessor devices and systems.— 1989.— N. 5 — P. 76.

Specific features of the 8-bit single-chip microcontroller families iMCS-48, iMCS-51 and K1816 are explained. Main technical data of auxiliary devices built into basic microcontroller chips are listed.

Номер подготовили:

Е. И. Бабич, Г. Г. Глушкова,
В. М. Ларионова
Корректор Е. М. Кучерявенко
Технический редактор
Г. И. Колосова
Адрес редакции журнала:
103051, Москва, Малый
Суваревский пер., д. 9А

Телефон: 208-73-23, 208-19-94
Сдано в набор 22.06.89. Т—23794
Подписано к печати 10.08.89.
Формат 84×108 1/16
Офсетная печать.
Усл. печ. л. 10,08. Уч.-изд. л. 14,3.
Тираж 108.400.
Заказ № 5947
Цена 1 р. 10 к.
Орган Государственного
комитета СССР

по вычислительной технике
и информатике

Набрано в ордена Трудового
Красного Знамени
Чеховском полиграфическом
комбинате Государственного
комитета СССР по делам
издательства, полиграфии и книжной
торговли. 142300, г. Чехов
Московской обл.

Отпечатано в Московской
типографии № 13 ПО «Периодика»
Государственного комитета
СССР по делам издательства,
полиграфии и книжной торговли.
107005, г. Москва, Денисовский
пер., 30.
Заказ 282.

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ

Издается с 1984 года

МП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 6 / 1989 МОСКВА

СОДЕРЖАНИЕ

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

Кузнецов Д. Н., Недоря А. Е., Тарасов Е. В., Филиппов В. Э.— Семейство процессоров для языков высокого уровня	2
Бобнов В. А., Кушарев В. Н., Свиридович В. С. и др.— Контроллер цифро-аналогового преобразователя KP588BG5	4
Бруевич Д. А., Воробьев Р. М., Куликов А. Г.— Одноплатное ОЗУ с контролем хранения данных	8
Михальски А., Свицер В.— Семейство графических устройств ввода-вывода	10
Синкевич Т.— Система отладки MSWP для 8-, 16- и 32-разрядных микропроцессоров	13
Казменко С. В.— Несколько этюдов разработки задач в среде электронных таблиц	16

В СТРАНАХ — ЧЛЕНАХ СЭВ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Пашенков С. В., Стерин М. Ф.— Мобильная файловая система с программным интерфейсом, совместимым с UNIX	21
Байдан И. Е., Гильен Н. Г. А., Гаджиев М. М., Депрес Д. С. Г.— Комплекс схемотехнического проектирования на микроЭВМ	24
Захаренков С. Я., Шехтер И. К., Захаренков А. С.— Измерительная система для аналоговых и аналого-цифровых интегральных микросхем	28
Семеренко В. П., Гуменюк Я. А.— Программа расчета сигнатур на ПЭВМ	29
Новиков В. В., Орехов А. В.— Интерактивная трассировка печатных плат на ДВКЗМ2	32

Прикладное программное обеспечение

Жарков Е. А., Ковальцов В. И., Максимов Г. М.— Кросс-система программирования 8- и 16-разрядных мп с унифицированным форматом объектных модулей	34
Пашенков С. В., Стерин М. Ф.— Разработка программ для микропроцессора K580ИК80 с использованием кросс-системы i85	36
Сафир М. Д., Сиянко В. М., Урсатий М. К.— Система отладки для секционированных микропроцессоров	39
Иванцов С. Я., Клим В. Я., Мамай В. И. и др.— Автоматизированная контролирующая система ПАКС-МК	40

Инструментальные программные средства

Балюк В. В., Кобзар С. П.— Диагностирование устройств на базе мп KP580ИК80А Мусагалиев Б. Ч.— Пульт отладки	42
Щелкунов Н. Н., Дианов А. П.— Средства и методы программирования ОМК	46

Средства отладки МП-систем

Музалевский И. В., Овчинников А. В., Розенштейн Э. П.— Построение арбитров на ПЗУ	49
Айтьян С. Х., Гуарян К. Р.— Инструментальные средства разработки экспертных систем на ПЭВМ	52
Белов В. А.— Построение экспертной системы средствами dBASE	55

Опыт разработки аппаратно-программных средств Экспертные системы

Коломыц В. Г., Королев И. В., Чернявский А. Д.— Система автоматизированного кроссового расчета	62
Черняк С. И., Табаткин В. М.— Индикатор шин микропроцессора K1810BM86	64

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ Локальные сети и средства межмашинной связи

Дьяченко А. М., Королев В. М., Симаненков С. И., Цейтлин Б. В.— Станция локальной сети «Электроника МС8301»	67
Новиков Ю. В.— Универсальный параллельный интерфейс для модульных микропроцессорных систем измерения, контроля и управления	69
Князев А. И., Шестимеров С. М.— Электронный диск с прямым доступом к памяти для ДВКЗМ	71
Гайдащенко Н. И.— Резервированный интерфейс для устройств связи с объектом	73

УЧЕБНЫЙ ЦЕНТР

Стацук И. П., Александрович А. П., Захаревич А. В.— Аппаратно-программный интерфейс для обмена информацией между ПЭВМ ЕС1840 и ДВКЗ	75
Лучук Д. А.— Синхронно-асинхронный адаптер с выходом на стык С2 для ЭВМ типа «Электроника 60»	77
Грушвицкий Р. И., Преображенский А. В., Симонов А. В., Яккер В. М.— Класс учебных ЭВМ в лабораторном практикуме «Операционные системы»	79
Статистическое ОЗУ K564PY2	83

УДК 621.3.049

Д. Н. Кузнецов, А. Е. Недоря, Е. В. Тарасов,
В. Э. Филиппов

КРОНОС: СЕМЕЙСТВО ПРОЦЕССОРОВ ДЛЯ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ

Кронос — общее название семейства 32-разрядных процессоров, предназначенных для создания микро- и мини-ЭВМ. Их архитектура ориентирована на поддержку языков программирования высокого уровня (Си, Модуль 2, Паскаль, Оккам). Язык Модуль 2, созданный Н. Виртом в 1975 г. на основе Паскаля и Модуль 1, унаследовал их лучшие свойства и был выбран базовым языком разработки.

К особенностям Модуль 2 относятся:

- модульность (введено понятие модуля и возможно его разбиение на определяющую и реализующую части);
- развитые структуры данных и управления;
- статическая типизация;
- наличие процедурных типов для динамического параметрирования процедуры внешними действиями;
- существование средств программирования низкого уровня, позволяющих ослабить жесткий контроль типов и отображение структурных данных в память.

Авторами использован опыт разработки ПЭВМ Lilith, выполненной под руководством Н. Вирта в Высшей технической школе в Цюрихе, однако многие решения приняты достаточно независимо. Так, при почти неизменном наборе команд доступа к значениям в оперативной памяти, арифметико-логических операций и структур управления существенно переработана архитектура взаимодействия процессов, прерываний, адресации и работы с внешними устройствами.

Широкое адресное пространство (до 4 млрд. слов) позволяет создавать виртуальную память для объектно-ориентированных моделей вычислений и разрабатывать системы искусственного интеллекта. Аппаратная поддержка прерываний по событиям, по синхронизации процессов, а также компактность кода программ обеспечивают успешное применение процессоров семейства Кронос в системах реального времени.

Архитектура процессоров Кронос отлична от традиционной:

- выражения вычисляются на быстром аппаратном стеке небольшой фиксированной глубины с сохранением его содержимого при вызове процедур-функций и переключении процессов;

- код и область данных любого процесса разделены (все программы и даже их отдельные части повторно входимы);

- даже в сегменте кода есть отказ от абсолютной адресации (таблицы смещения начала процедур упрощают их вызовы);

- развитые виды адресации отражают понятия современных языков программирования (адресация локальных, глобальных, внешних объектов и объектов статически вложенных процедур);

- запатентованный метод межмодульной адресации позволяет организовать динамическую загрузку-связывание-исполнение программ;

- наличие развитого аппарата работы с мультисзначениями.

Введем некоторые понятия. Программа, исполняемая в текущий момент процессором, называется процессом. Он состоит из следующих компонент: таблицы загруженных в

данный момент сегментов кода отдельно скомпилированных модулей [DFT], глобальной области данных [G], сегмента кода [F], области строковых констант [STRINGS], процедурного стека (П-стека) [P].

В квадратных скобках указаны базовые регистры, относительно которых происходит адресация внутри компонента. DFT (data frame table) поддерживает динамическое связывание отдельно скомпилированных модулей. П-стек размечается тремя указателями: L — начало локальной области данных исполняемой процедуры, S — начало свободной части П-стека, H — ограничитель П-стека (предел увеличения S).

Переполнение П-стека (перекрытие S и H) отслеживается аппаратно с возбуждением соответствующего прерывания.

Способы адресации архитектуры Кронос: локальный, глобальный, внешний, промежуточный и косвенный.

Промежуточный способ — обращение к нелокальным переменным внутри процедуры и проход по статической цепочке процедур.

Значения всех структурных переменных (массивов и записей) представлены косвенно с помощью ссылки на начало выделенного им участка памяти. Структурные компоненты таких значений расположены внутри. Таким образом, удалось ограничить максимальный размер статички процедуры до 256 слов, т. е. процедура в терминах Модуль 2 не может содержать более 256 локальных переменных.

Выражения вычисляются на арифметическом стеке (А-стеке). Компилятор статически отслеживает пополнение А-стека и в случае необходимости может сохранять его содержимое в памяти и восстанавливать.

М-код — поток байтов (по коду операции всегда однозначно известно, сколько байтов непосредственных операндов следует за ним). Большинство команд — однобайтовые. Извлечение команд и операндов из потока производится с одновременным продвижением вперед счетчика команд РС.

М-код имеет четыре модификации длины непосредственного операнда: полубайтовый операнд (последние 4 бит команды интерпретируются как операнд), байтовый, двух- и четырехбайтовый операнды (представляемые следующими за кодом команды байтом, двумя или четырьмя байтами соответственно). Система команд построена так, что наиболее часто производимые операции имеют более короткий код, что увеличивает скорость и уменьшает его размер. Так, загрузку на стек выражений чисел 0, 1, ..., 15 выполняют команды LI0, LI1, ..., LI15 — Load Immediate, занимающие вместе с операндом один байт кода, в то время как для загрузки на стек числа 153 используется уже команда LIB — Load Immediate Byte, занимающая вместе с операндом два байта кода.

Процедуры представлены своим номером из диапазона 0... 255. Кодовый сегмент содержит таблицу смещений процедур относительно его начала. Поэтому команды вызова процедур также имеют длину один (для процедур с номерами 0... 15) или два байта.

Механизм передачи параметров процедурам различен. Если у процедуры менее семи параметров, они передаются через арифметический стек (расходы на обработку параметров в точке вызова сокращаются). Все арифметико-логические операции работают над одним или двумя верхними элементами стека и помещают результат на место аргументов.

В начале П-стека всегда хранится так называемый дескриптор процесса, представленный копией всех регистров в момент последнего переключения на процесс. Команда

переключения процессов сохраняет текущее состояние регистров в дескрипторе задерживаемого процесса, извлекает состояние регистров возобновляемого процесса и продолжает работу в возобновленном процессе. Этот аналог переключения сопрограмм — мощное средство для организации мультипрограммной работы и синхронизации.

Все прерывания обрабатываются, как переключения процессов, указатели которых находятся в элементе массива векторов прерываний, соответствующем номеру прерывания.

Семейство процессоров Кронос 2.X включает три разработки (2.2, 2.5, 2.6). У всех процессоров одна система команд, различие — во внутреннем функциональном устройстве, быстродействии и конструктивном исполнении.

Логику функционирования всех блоков процессора реализует блок микропрограммного управления. Две шины данных объединяют арифметико-логическое устройство, блок регистров, быстрый аппаратный стек на семь слов, устройство выборки команд и буфер связи с шиной ввода-вывода.

С помощью двухшинной внутренней структуры процессора выполняются бинарные операции на стеке (сложение, вычитание, логические И, ИЛИ и т. д.) за один такт. Это существенно повышает производительность процессора на основном наборе команд. Микропрограммное управление упрощает устройство процессоров и позволяет реализовать сложные команды.

Все узлы процессоров выполнены на отечественных микросхемах (ТТЛ и ТТЛШ) серий К155, К531, К1802, К1804, К589, К556.

Кронос 2.2 — первая реализация изложенных архитектурных концепций. Процессор полностью совместим со всеми устройствами, поддерживающими протокол обмена по шине МПИ, и оперирует 32-разрядными словами. Доступ к слову данных в памяти происходит за два обращения по шине МПИ, так как шина — 16-разрядная, АЛУ — 20-разрядное. Адресная арифметика обрабатывается за один такт, данные — за два такта. Такая внутренняя организация позволила разместить весь процессор на одной плате при существующей элементной базе.

Объем прямо адресуемой оперативной памяти процессора — до 4 Мбайт, тактовая частота — 4 МГц, производительность в секунду: 600 тыс. операций над стеком и 250 тыс. команд обмена с памятью.

Кронос 2.5 — полностью 32-разрядный (две платы в конструктиве Intel). Интерфейс с внешними устройствами реализуется через шину Multibus 1. Существенное отличие — локальная память объемом 0,5... 2 Мбайт (в зависимости от применяемых микросхем). Тактовая частота процессора — 3 МГц, производительность в секунду: 1,5 млн. простых операций над стеком и 1 млн. команд обмена с памятью.

Кронос 2.6 — дальнейшее развитие процессора, который может быть встроен в отдельную мини-ЭВМ или в мультипроцессорный комплекс. Он выполнен в конструктиве Евромеханика. Размеры плат 233,3×220 мм (Е2).

В минимальный комплект включены: платы обрабатываемого тракта (АЛУ, стек, регистры), микропрограммного управления, локальной памяти (0,5—2 Мбайт), адаптера шины ввода-вывода. Все устройства объединены локальной синхронной 32-разрядной шиной. Сам процессор не зависит от конкретной шины ввода-вывода, настройка производится с помощью соответствующего адаптера. К локальной шине можно подключить платы памяти, адаптера межпроцессорной связи, контроллера локальной сети и накопителя на магнитных дисках, памяти кода (при разделении плат кода и данных), bitmap-дисплея, арифметического вычислителя и других устройств, расширяющих возможности процессора.

Тактовая частота процессора — 3 МГц, скорость в секунду: 2 млн. операций над стеком и 1,5 млн. команд обмена с памятью.

ЭВМ с процессорами Кронос 2.2, Кронос 2.5 и Кронос 2.6 работают под управлением ОС Excelsior, предназначенной для решения широкого круга задач. Принципы

создания: открытость, модульность, интегрируемость результатов и удобство интерфейса. Под открытой понимается система, в которой:

любое созданное программистом ПО автоматически наращивает программные средства и входит в их состав на равных правах с базовым ПО;

для работы любого ПО необходимы только те программные и аппаратные средства, которые задействованы в нем непосредственно. Поясним на примере. Если программист создает программу для встроенного применения, не предполагающую работу с файлами и дисковой памятью и не нуждающуюся в существующей файловой системе, ему гарантируется возможность получить продукт (в который не входит файловая система и ее приложения), способный функционировать как на рабочем месте, так и в целевых условиях.

Модульность ПО достигается прежде всего свойствами языка Модула и поддерживает динамическую загрузку задач, с помощью которой из цикла редактирование — компиляция — запуск можно исключить этап сборки задачи и значительно ускорить этот цикл. Кроме того, динамическая загрузка избавляет от необходимости хранить многочисленные копии кода часто используемых модулей, например библиотек.

Отсутствие многочисленных копий кода позволяет существенно экономить внешнюю память (в сравнении с «линкованием» со стандартными библиотеками традиционных систем).

Пример внутренней интегрируемости — все программные средства ОС Excelsior, построенной в виде модульной расслоенной иерархической системы. Опыт эксплуатации машин Кронос как автоматизированных рабочих мест программиста в различных коллективах показал, что большая часть создаваемого программистами ПО выглядит не только как самостоятельный программный продукт, но и как набор библиотек, реализующих абстрактные типы данных. Эти библиотеки можно использовать в другом ПО, они постоянно пополняют набор «строительных материалов», из которых программисты Кроноса складывают свои программы.

Удобство интерфейсов — главнейший принцип из положенных разработчиками во все разрабатываемые системы. Необходимы самонастраивающиеся средства, максимально полно использующие возможности каждого конкретного вида оборудования, обслуживающего рабочее место программиста. Недопустимо просто минимизировать возможности терминалов, клавиатур и других устройств, так как это отрицательно сказывается на удобстве работы программиста. В ОС Excelsior эти проблемы решаются с помощью техники абстрактных типов данных. Вся работа с конкретными устройствами инкапсулирована в различные реализации одного и того же определяющего модуля, поддерживающего интерфейс всех программных средств с устройствами такого класса. Поэтому сама система и прикладное ПО оперируют с устройствами только в терминах такого определения. Этим дополнительно достигается полное единообразие работы с различными программными средствами (даже еще не знакомыми пользователю). В любой момент работы одним для всех программных средств способом можно получить толковую и понятную подсказку на родном для программиста языке.

Все программные средства ОС Excelsior ориентированы на мультипроцессорное, многозадачное и многопользовательское применение. Благодаря модульной архитектуре спектра применения процессоров Кронос широк — от встроенных систем реального времени до супермини-ЭВМ. Самое замечательное свойство — легкость программирования (programmability) — делает Кронос незаменимым в приложениях, требующих постоянного развития и изменения ПО.

Телефон 35-41-10, Новосибирск

Статья поступила 9.12.88

1. Wirth N. Programming in Modula 2. Third, Corrected Edition, Springer-Verlag.— Berlin, Heidelberg, New-York, 1985.
2. Wirth N. A Fast and Compact Compiler for Modula 2. ETH.— Zurich, 1985.
3. Кузнецов Д. Н., Недоря А. Е., Осипов А. Е., Тарасов Е. В. Процессор Кронос в мультипроцессорной системе // Вычислительные системы и программное обеспечение. — Новосибирск, 1986. — С. 13—19.
4. Кузнецов Д. Н., Недоря А. Е. Проектирование таблиц символов для языков со сложными правилами видимости // Методы трансляции и конструирования программ. — Новосибирск, 1984. — С. 153—158.
5. Васекин В. А., Кузнецов Д. Н., Недоря А. Е., Тарасов Е. В. Кронос: аппаратная и программная поддержка языков высокого уровня. // Тез. докл. Всероссийского методического совещания-семинара по обмену опытом использования ЭВМ в учебной и научно-исследовательской работе студентов. — Новосибирск, 1986. — С. 92—94.
6. Кузнецов Д. Н., Недоря А. Е. Симфайлы как интерфейс операционной системы // Информатика. Технологические аспекты. — Новосибирск, 1987. — С. 68—75.
7. Кузнецов Д. Н., Недоря А. Е., Тарасов Е. В., Филиппов В. Э. Кронос — автоматизированное ра-

Рабочая группа по языку программирования Модуля 2 при Совете АН СССР по автоматизации научных исследований и Центр научно-технической деятельности, исследований и социальных инициатив при АН СССР предполагают начать в 1989 г. оказание следующих видов услуг для министерств, ведомств, предприятий, учреждений, кооперативов и отдельных граждан:

1. Курсы по программированию на Модуля 2.
2. Абонементное обслуживание по материалам, связанным с языками программирования Модуля 2 и Oberon: рассылка подборок ксерокопий зарубежных статей из новых поступлений в ГПНТБ (периодичность — раз в два месяца, ориентировочный объем 300... 400 страниц в год).
3. Изготовление 32-битных процессоров «Кронос П2.2» (в основном из комплектующих заказчика).
4. Установка системного программного обеспечения для «Кронос П2.2» на комплекс заказчика, его доработка и сопровождение.
5. Разработка прикладного и системного ПО на языке Модуля 2. Просим сообщить ориентировочную потребность в указанных видах услуг.

117333, Москва, ул. Вавилова, 38, ИОФАН;
тел. 234-37-19, Сагателян Д.М.

УДК 681.326.35.77

В. А. Бобков, В. Н. Кушарев, В. С. Свиридович, Б. Н. Чернуха, Д. Н. Черняковский

КОНТРОЛЛЕР ЦИФРО-АНАЛОГОВОГО ПРЕОБРАЗОВАТЕЛЯ КР588ВГ5

Контроллер ЦАП (КЦАП) состоит из следующих функциональных блоков (рис. 1):

управления обменом по каналу МПИ (БУОК) для взаимодействия контроллера с ЦП и формирования сигналов управления магистральными приемопередатчиками;

селекции адреса (БСА), формирующего сигналы выбора внешних регистров;

кварцевого генератора (ГК);

управления обменом в режиме прямого доступа к памяти (БУОПДП), который при наличии в системе контроллера прямого доступа к памяти (КПДП) организует запись или чтение из ОЗУ ЦП в ОЗУ ЦАП;

буфера адреса (БА) для хранения адреса ячейки ОЗУ ЦАП;

буфера канала (БК), развязывающего контроллер с информационным каналом МПИ;

5-разрядного суммирующего счетчика

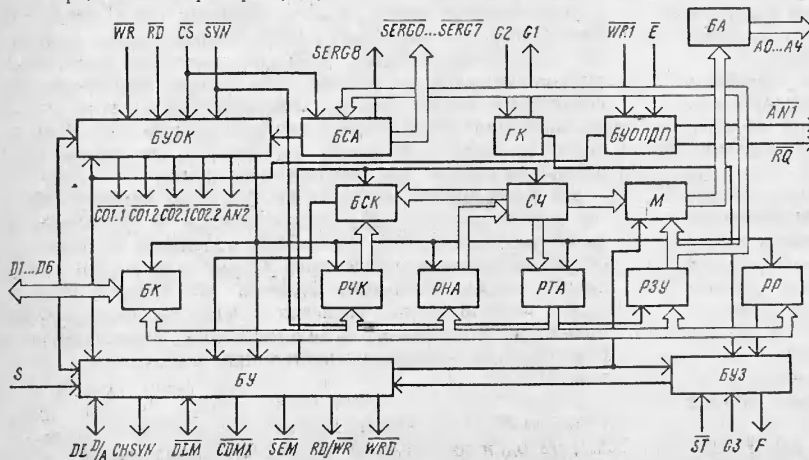


Рис. 1. Функциональная схема КЦАП

(СЧ), формирующего адрес ячейки ОЗУ ЦАП в режиме ПДП и циклах ЦАП; 5-разрядных регистров: числа каналов (РЧК), начального адреса (РНА), текущего адреса (РТА).

6-разрядного регистра запоминающего устройства (РЗУ) для хранения адреса ячейки ОЗУ ЦАП или когда внешнего регистра ЦАП с разрядами следующего назначения:

РЗУ1...РЗУ3 задают адрес внешнего регистра;

РЗУ4 выбирает одну из двух групп внешних регистров;

РЗУ5 разрешает (РЗУ5=0) или запрещает (РЗУ5=1) работу с внешними регистрами;

РЗУ6 определяет источник информации в цикле ЦАП, т. е. при РЗУ6=0 выбирается ОЗУ ЦАП, где содержится разряды РЗУ1...РЗУ5 задает адрес ячейки ОЗУ, а при РЗУ6=1 выбираются внешние регистры ЦАП.

5-разрядного регистра режима (РР), определяющего работу контроллера и имеющего разряды следующего назначения:

РР1 разрешает (РР1=1) или запрещает (РР1=0) работу контроллера по циклу ЦАП;

РР2...РР4 содержат код коэффициента деления внутренней тактовой частоты в соответствии с табл. 1;

РР5 определяет условия получения внутренней тактовой частоты, которая при РР=0 поступает с внутреннего ГК, а при РР=1 — с внешнего генератора;

Таблица 1

Коэффициенты деления внутренней тактовой частоты

K _{дел}	Состояние разряда PP		
	PP4	PP3	PP2
1	0	0	0
5	0	0	1
10	0	1	0
20	0	1	1
40	1	0	0
100	1	0	1
200	1	1	0
400	1	1	1

сравнения номера обслуживаемого канала (БСК) с номером канала, который будет обслуживаться в следующем цикле;

мультиплексора (М), коммутирующего сигналы, поступающие в БА с СЧ и РЗУ;

управления (БУ) ЦАП; управления запуском (БУЗ) цикла ЦАП.

Условное графическое обозначение микросхемы приведено на рис. 2, назначение выводов показано в табл. 2, основные технические характеристики приведены ниже.

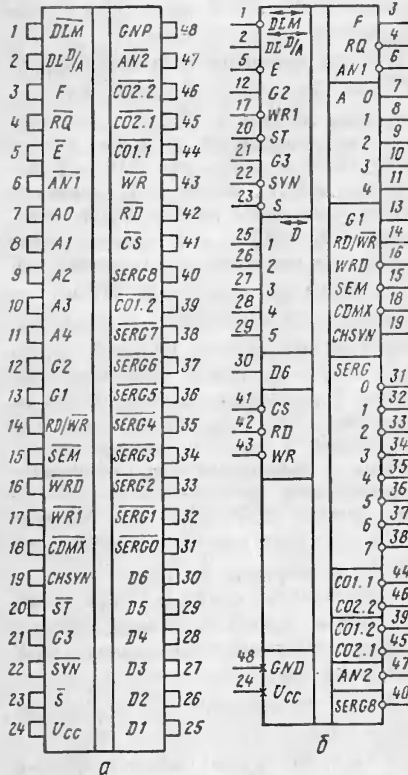


Рис. 2. Условное графическое обозначение микросхемы КЦАП по порядку расположения (а) и функциональному назначению (б) выводов

Таблица 2

Назначение выводов КЦАП КР588ВГ5

Вывод	Тип	Обозначение	Назначение
1	Вход-выход	DLM	Задержка ЗУ
2	»	DLD/A	Задержка преобразователя
3	Выход	F	Частота делителя
4	»	RQ	Запрос ПДП
5	Вход	E	Разрешение ПДП
6	Выход	AN1	Ответ прямого доступа
7...11	»	A0...A4	Каналы адреса
12	Вход	G2	Вход ГК
13	Выход	G1	Выход ГК
14	»	RD/WR	Чтение-запись
15	»	SEM	Выборка ЗУ
16	»	WRD	Запись данных
17	Вход	WR1	Запись 1
18	Выход	CDMX	Строб демультимплексора
19	»	CHSYN	Контроль синхронизации
20	Вход	ST	Внешний запуск
21	»	G3	Внешний генератор
22	»	SYN	Синхронизация обмена
23	»	S	Начальная установка
24	—	U _{CC}	Напряжение источника питания
25...29	Вход-выход	D1...D5	Каналы данных
30	Вход	D6	Канал данных
31...38	Выход	SERG0...SERG7	Выборка регистров
39	»	C01.2	Управление МПП
40	»	SERG8	Выборка группы внешних регистров
41	Вход	CS	Выборка кристалла
42	»	RD	Чтение
43	»	WR	Запись
44...46	Выход	C01.1, C02.1, C02.2	Управление МПП
47	»	AN2	Ответ
48	—	GND	Общий

Технические характеристики КЦАП, T = -60...+85°С

Напряжение питания U_{CC}, В. 5 ± 10 %
 Выходной ток низкого уровня при выходном напряжении 0,4 В, мА, не менее 0,8
 Выходной ток высокого уровня при выходном напряжении -0,4 В, мА, не менее 0,4
 Ток утечки по входам, мкА, не более 20
 Время записи-чтения информации при C_L ≤ 100 пФ, U_{CC} = 5,0 В, U_{IL} = -0,4 В, U_{IK} = U_{CC} - 0,4 В, ис, не более 1000

Устройство преобразования цифровых сигналов (рис. 3) включает КЦАП КР588ВГ5, СА КР588ВТ1, магистральные приемопередатчики (МПП) КР588ВА1, ОЗУ ЦАП, буферный регистр (БРГ), ЦАП, 32-канальный демультимплексор (ДМ) аналогового сигнала. Приемопередатчики предназначены для развязки информационного канала МПИ и электрического согласования с КЦАП и ОЗУ ЦАП. СА согласует контроллер с МПИ. ОЗУ ЦАП необходимо для хранения данных, которые в ЦАП переписываются в БРГ, а затем преобразуются в аналоговый сигнал. ДМ коммутирует аналоговый сигнал, поступающий с ЦАП, на один из 32 каналов, номер которого совпадает с адресом ОЗУ ЦАП.

Режимы работы КЦАП

Обмен по каналу МПИ. ЦП в стандартных циклах ВЫВОД записывает информацию во внутренние регистры РР, РЧК, РНА, в ОЗУ ЦАП, а в циклах ВВОД считывает информацию из внутренних регистров РР, РЧК, РНА, РТА, ОЗУ ЦАП.

Запись в ОЗУ ЦАП. Запись происходит в режиме ПДП при наличии в системе КПДП.

Обслуживание ЦАП. Данные из ОЗУ ЦАП переписываются в БРГ, преобразуются ЦАП и поступают в ДМ для коммутации на канал в соответствии с A0...A4.

Микросхема КР588ВГ5 устанавливается в начальное состояние подачей отрицательного импульса длительностью не менее 500 нс на вход S, по которому сигнал C01.1 переходит в 0, сигналы C01.2, C02.1, C02.2 — в 1, а все другие сигналы принимают значения, соответствующие условному графическому обозначению (см. рис. 2).

Алгоритм записи-чтения данных во внутренние регистры контроллера одинаков для всех регистров. Их адреса приведены в табл. 3, временная диаграмма записи данных в регистры РР, РЧК, РНА — на рис. 4.

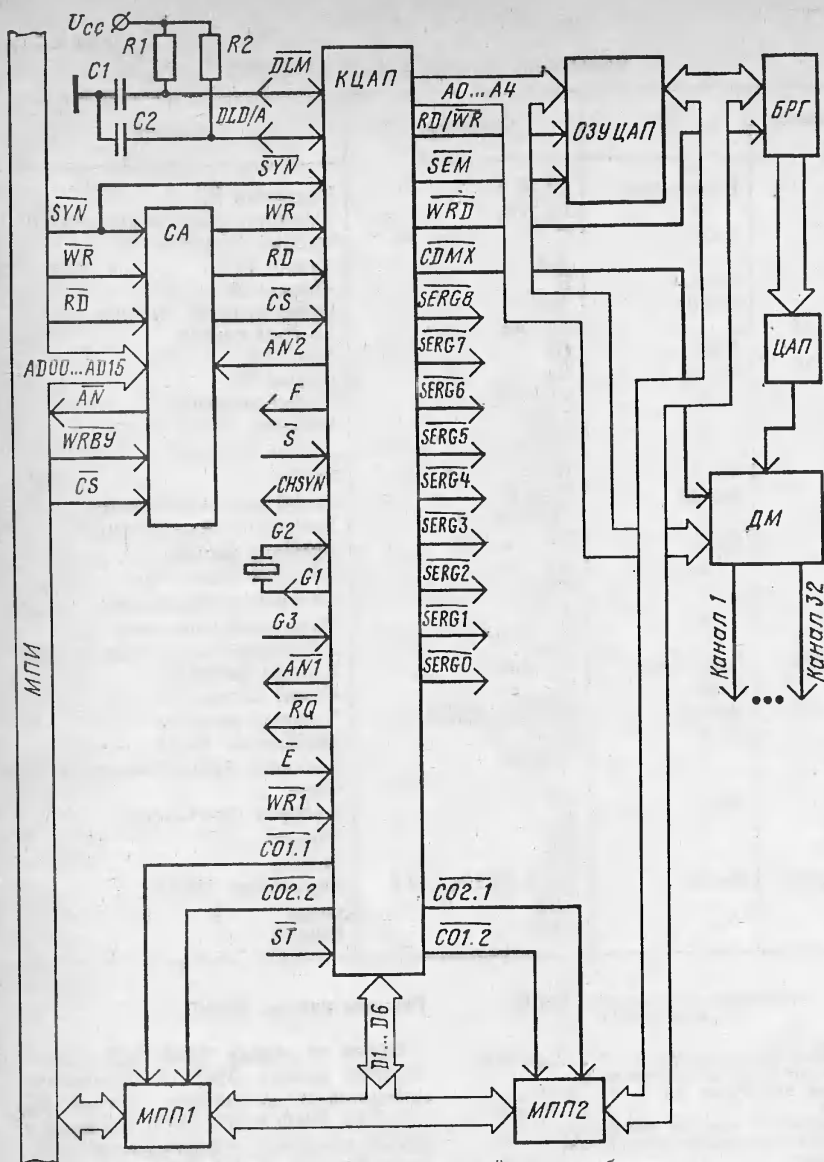


Рис. 3. Структурная схема устройства преобразования цифровых сигналов

После начальной установки КЦАП МПП1 переходит в режим передачи информации из МПИ к контроллеру. ЦП задает на информационном канале AD00...AD15 адрес, устанавливает в 0 сигналы \overline{CS} , \overline{WRBY} , \overline{SYN} . Получив

Таблица 3

Адреса внутренних регистров КЦАП

Регистр	Адрес					
	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁
РТА	1	1	0	0	0	0
РР	1	1	0	0	0	1
РНА	1	1	0	0	1	0
РЧК	1	1	0	0	1	1

сигнал $\overline{SYN}=0$, СА сравнивает данные на выводах AD04...AD12 с данными на выводах A4...A12 и при их совпадении активируется, устанавливает сигнал \overline{CS} в 0, по которому в КЦАП в соответствии с табл. 3 выбирается внутренний регистр. Затем КЦАП переводит в 1 сигнал C01.1, закрывая МПП1. ЦП снимает адрес с информационного канала, переводит сигналы \overline{CS} , \overline{WRBY} в 1 и устанавливает на AD00...AD15 данные, сопровождая их сигналом $\overline{WR}=0$, по которому СА задает $\overline{WR}=0$. Получив сигнал $\overline{WR}=0$, КЦАП открывает МПП1 на передачу данных из канала МПИ, определяет C01.1=0, записывает данные с МПП1 в выбранный регистр, обеспечивает $\overline{AN2}=0$.

По сигналу $\overline{AN2}=0$ СА устанавливает сигнал $\overline{AN}=0$, ЦП — $\overline{WR}=1$, СА — $\overline{WR}=1$, КЦАП — $\overline{AN2}=1$, СА — $\overline{AN}=1$, ЦП — $\overline{SYN}=1$, КЦАП — C01.1=0. На этом запись данных во внутренний регистр КЦАП завершается.

Временная диаграмма чтения внутренних регистров РР, РЧК, РНА, РТА приведена на рис. 5. Адресная часть цикла чтения аналогична циклу записи, за исключением того, что сигнал $\overline{WRBY}=1$. ЦП после снятия адреса с информационного канала AD00...AD15 задает $\overline{RD}=0$, СА переводит свой сигнал \overline{RD} в 0, КЦАП — сигнал C02.1 в 0, открывая МПП1 на передачу информации в МПИ, устанавливая на D1...D5 данные выбранного регистра и подтверждая их достоверность равенствами $\overline{AN2}=0$ и $\overline{AN}=0$. ЦП по сигналу $\overline{AN}=0$ читает данные с информационного канала, задает $\overline{RD}=1$, СА — $\overline{AN}=1$ и $\overline{RD}=1$, КЦАП — $\overline{AN2}=1$ и C02.1=1, закрывая МПП1, ЦП — $\overline{SYN}=1$, по которому КЦАП переводит C01.1 в 0, а СА — сигнал \overline{CS} в 1. Чтение данных внутреннего регистра КЦАП на этом заканчивается.

Временная диаграмма записи данных в ОЗУ ЦАП приведена на рис. 6. ЦП устанавливает на информационном канале AD00...AD15 адрес СА и задает равенство нулю сигналов \overline{CS} , \overline{WRBY} , \overline{SYN} . СА активируется и обеспечивает $\overline{CS}=0$. При $\overline{PZY6}=0$ КЦАП, получив сигналы $\overline{SYN}=0$ и $\overline{CS}=0$, устанавливает на выводах A0...A4 адрес ячейки ОЗУ ЦАП и определяет C01.1=1, закрывая МПП1. Затем ЦП снимает с информационного канала адрес, переводит \overline{CS} , \overline{WRBY} в 1, устанавливает данные на информационном канале, сопровождая их равенством $\overline{WR}=0$, по которому СА задает $\overline{WR}=0$, КЦАП — C01.1=0, $\overline{RD}/\overline{WR}=0$, $\overline{DLM}=0$, открывая МПП1 на прием информации из МПИ. Длительность сигнала \overline{DLM} определяется параметрами RC-цепи, подключенной к данному выводу микросхемы, и выбирается в соответствии с необходимым временем на запись-чтение ячейки ОЗУ ЦАП. По сигналу $\overline{DLM}=0$ КЦАП обеспечивает C01.2=0, $\overline{SEM}=0$, открывая МПП2 на передачу и запись данных в ОЗУ ЦАП; по сигналу $\overline{DLM}=1$ КЦАП — $\overline{AN2}=0$, $\overline{SEM}=1$, информируя об окончании записи в ОЗУ ЦАП. СА при $\overline{AN2}=0$ переводит сигнал \overline{AN} в 0, по которому ЦП определяет $\overline{WR}=1$, СА — $\overline{WR}=1$, $\overline{AN}=1$, КЦАП — C01.1=1, $\overline{RD}/\overline{WR}=1$, $\overline{AN2}=1$, C01.2=1, ЦП — $\overline{SYN}=1$, СА — $\overline{CS}=1$, КЦАП — C01.1=1 и снимает адрес ОЗУ ЦАП с A0...A4. Запись данных в ОЗУ ЦАП выполнена.

Временная диаграмма чтения данных из ОЗУ ЦАП приведена на рис. 7. Ад-

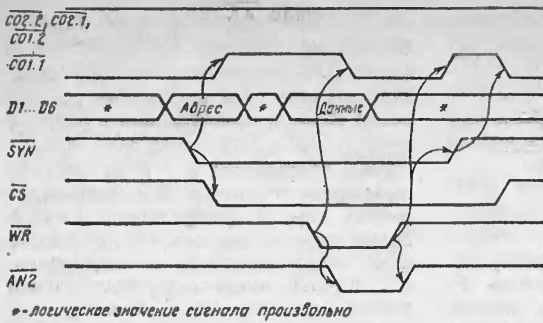


Рис. 4. Временная диаграмма записи данных во внутренние регистры

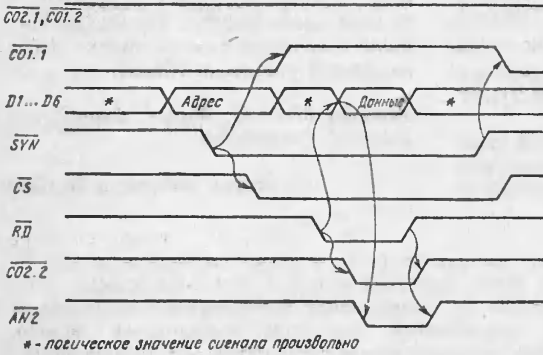


Рис. 5. Временная диаграмма чтения данных из внутренних регистров

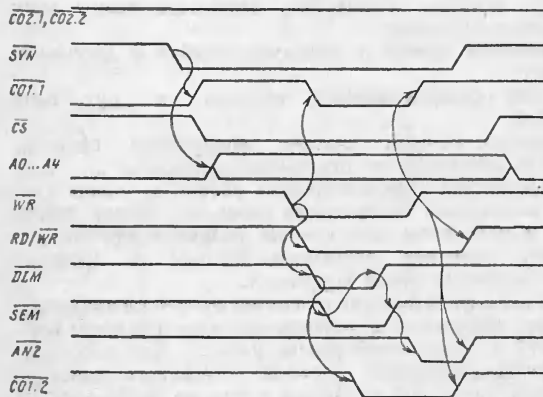


Рис. 6. Временная диаграмма записи данных в ОЗУ ЦАП

решая часть цикла чтения аналогична циклу записи, но при этом $WRBY=1$. ЦП после снятия адреса СА с информационного канала устанавливает $RD=0$, СА — $RD=0$, КЦАП — $CS0.1=0$, $CS0.2=0$, $DLM=0$, $SEM=0$ и открывает МПП1 и МПП2 на передачу данных к МПИ. ОЗУ ЦАП, получив сигналы $SEM=0$ и $RD/WR=1$, определяет данные в соответствии с адресом $A0...A4$. По сигналу $DLM=1$ КЦАП пере-

длит сигнал $AN2$ в 0, информируя СА о наличии данных в каналах МПП1, МПП2. СА в свою очередь обеспечивает $AN=0$, а ЦП читает данные с информационного канала, затем задает $RD=1$.

Далее СА устанавливает $RD=1$, $AN=1$, КЦАП — $SEM=1$, $AN2=1$, $CS0.1=1$, $CS0.2=1$. ОЗУ ЦАП снимает данные. ЦП по сигналу $AN=1$ определяет равенство $SYN=1$, СА — равенство

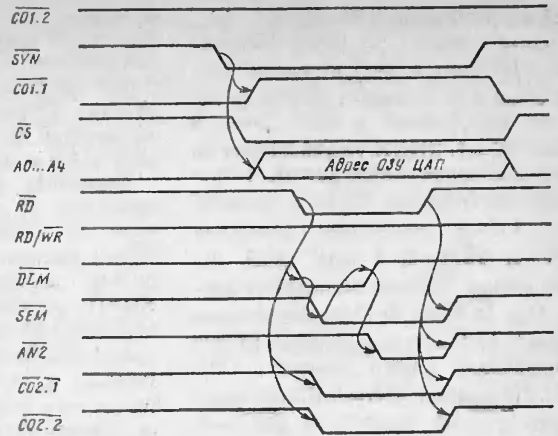


Рис. 7. Временная диаграмма чтения данных из ОЗУ ЦАП

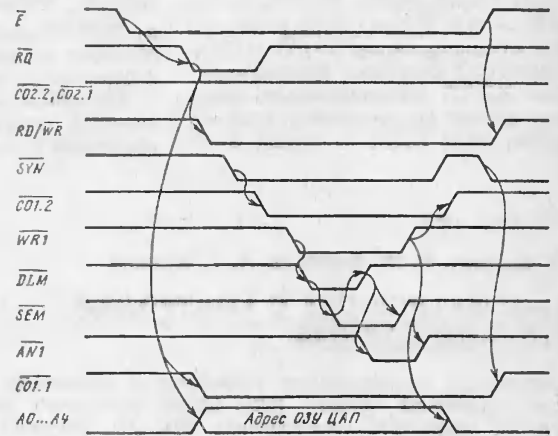


Рис. 8. Временная диаграмма записи данных в ОЗУ ЦАП в режиме ПДП

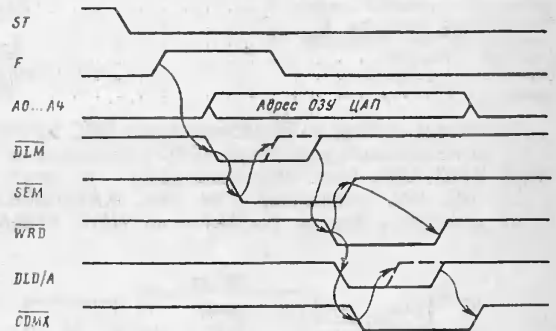


Рис. 9. Временная диаграмма цикла ЦАП

$CS=1$. КЦАП снимает адрес с $A0...A4$ и переводит сигнал $CS0.1$ в 0. Чтение данных из ОЗУ ЦАП закончено.

Временная диаграмма записи данных в ОЗУ ЦАП в режиме ПДП приведена на рис. 8. КЦАП после окончания текущего цикла ЦАП по сигналу $E=0$ с КПДП задает в СЧ из РНА адрес, начиная с которого данные будут записываться в ОЗУ ЦАП.

КПДП при $RQ=0$ начинает цикл

ПДП и устанавливает на выводах А0...А4 адрес ячейки ОЗУ ЦАП. Получив сигнал $\overline{SYN}=0$, КЦАП переводит сигнал $\overline{CO1.2}$ в 0, открывая МПП2 на передачу информации в ОЗУ ЦАП, и задает $\overline{RQ}=1$. КПДП устанавливает на информационном канале данные, сопровождая их сигналом $\overline{WR1}=0$, по которому КЦАП определяет равенства $\overline{DLM}=0$, $\overline{SEM}=0$, а ОЗУ ЦАП, получив сигнал $\overline{SEM}=0$, записывает данные. При $\overline{DLM}=1$ КЦАП обеспечивает $\overline{SEM}=1$, $\overline{ANI}=0$ и информирует КПДП об окончании записи данных в ОЗУ ЦАП. По сигналу $\overline{ANI}=0$ КПДП определяет $\overline{WR1}=1$, КЦАП — $\overline{CO1.2}=1$, $\overline{ANI}=1$, снимает адрес ячейки ОЗУ ЦАП с выводов А0...А4. Получив $\overline{ANI}=1$, КПДП задает $\overline{SYN}=1$, $\overline{E}=1$, КЦАП — $\overline{RD}/\overline{WR}=1$. Запись информации по одному адресу в ОЗУ ЦАП в режиме ПДП считается выполненной.

Если КПДП последовательно записывает данные по нескольким адресам в одном цикле ПДП, то сигнал $\overline{E}=0$

в течение всего цикла. Содержимое СЧ КЦАП инкрементируется после записи отдельного слова в ОЗУ ЦАП. Таким образом, каждый раз с приходом сигнала $\overline{SYN}=0$ КЦАП устанавливает на выводах А0...А4 новый адрес ячейки ОЗУ ЦАП и цикл повторяется.

Временная диаграмма цикла ЦАП приведена на рис. 9. Цикл начинается, если на входе КЦАП есть сигнал $\overline{ST}=0$. По положительному фронту импульса, поступающего с выхода F, КЦАП устанавливает адрес ячейки ОЗУ ЦАП на выводах А0...А4 и обеспечивает $\overline{DLM}=0$, $\overline{SEM}=0$. По сигналу $\overline{DLM}=1$ КЦАП переводит \overline{WRD} в 0. По фронту этого сигнала данные, поступающие из ОЗУ ЦАП, записываются в БРГ, после чего КЦАП определяет равенства $\overline{DLD}/\overline{A}=0$, $\overline{CDMX}=0$, $\overline{SEM}=1$, $\overline{WRD}=1$. ОЗУ ЦАП по сигналу $\overline{SEM}=1$ снимает данные, ЦАП преобразует данные, поступающие с БРГ, и передает их в ДМ.

Длительность сигнала $\overline{DLD}/\overline{A}$ определяется параметрами RC-цепи, подключенной к данному выводу микросхе-

мы, и соответствует времени, необходимому на преобразование данных в ЦАП. ДМ, получив сигнал $\overline{CDMX}=0$, записывает в свой внутренний регистр номер канала, совпадающий с адресом ячейки ОЗУ. При $\overline{DLD}/\overline{A}=1$ сигнал \overline{CDMX} переводится в 1 и по нему на выбранном канале в ДМ устанавливаются данные, поступающие с ЦАП. Затем адрес с выводов А0...А4 ячейки ОЗУ ЦАП снимается и содержимое СЧ КЦАП инкрементируется. Цикл повторяется с приходом нового фронта импульса на вывод F. Микросхема может работать в цикле ЦАП с 16 внешними регистрами, каждый из которых работает на свой ЦАП.

На временных диаграммах работы микросхемы КР588ВГ5 приведены сигналы, непосредственно участвующие в данном цикле работы. Остальные сигналы сохраняют свое состояние после начальной установки КЦАП.

Телефон 77-96-53, Минск, Свиридович Валерий Степанович

Статья поступила 28.12.88

УДК 681.327.2—185.4

Д. А. Бруевич, Р. М. Воробьев, А. Г. Куликов

ОДНОПЛАТНОЕ ОЗУ С КОНТРОЛЕМ ХРАНЕНИЯ ДАННЫХ

Оперативное запоминающее устройство с самоконтролем и коррекцией ошибок (ОЗУ-СКМ) выполнено на полуплате микроЭВМ «Электроника 60» (см. рисунок).

Технические характеристики ОЗУ-СКМ

Емкость, Кбайт	128
Длина слова, бит	8,16
Длительность цикла, мкс	1,2
Напряжение источника питания, В	$5 \pm 0,25$
Потребляемая мощность, Вт	5
Диапазон рабочих температур, °С	5...50
Внешний интерфейс	МПП1
Габаритные размеры, мм	135×240
Масса, г	200

Накопитель состоит из 26 динамических БИС КР565РУ5. Блок магистральных усилителей (БМУ) выполнен на семи ИМС КР531АП2, блок коррекции (БК) — на двух СИС КР1801ВП1-039, контроллер — на СИС КА1515ХМ1-034, схема опознавания адреса (СхОА) — на ИМС К555КП15,

генератор импульсов (ГИ) и схема индикации (СхИ) — на двух ИМС К155ЛН1 и одной ИМС К155ЛЛ1.

В отличие от выпускаемых отечественной промышленностью аналогичных устройств оперативная память ОЗУ-СКМ обладает рядом дополнительных функциональных возможностей:

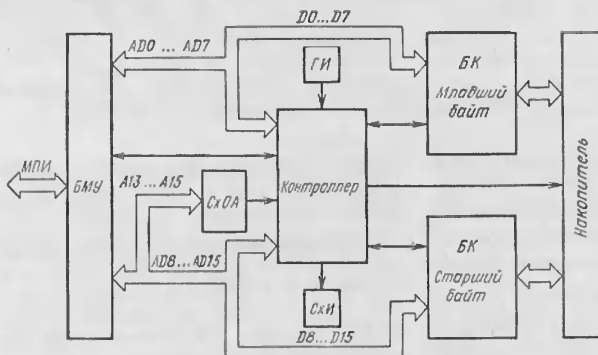
- аппаратной записи в каждую ячейку накопителя кодовых слов после включения питания;
- последовательного считывания всех ячеек в процессе работы для регенерации и обнаружения ошибок;
- выдачи вектора прерывания, соответствующего типу обнаруженной ошибки;
- запоминания адреса и синдрома ошибки в доступных регистрах;
- создания резервированной системы из двух плат ОЗУ-СКМ.

В диагностическом режиме аппаратные средства ОЗУ-СКМ обеспечивают отключение коррекции для проверки накопителя информационных разрядов; чтение данных из накопителя контрольных разрядов; запрет записи данных в накопитель контрольных разрядов для преднамеренного внесения одиночной ошибки и проверки работоспособности блока коррекции.

После включения питания генератор импульсов инициирует циклы обращения к накопителю для перевода БИС КР565РУ5 в нормальный режим работы. Для последующего функционирования системы встроенного контроля по сигналу начальной установки в течение 65536 рабочих циклов в каждую ячейку ОЗУ записывается ее адрес. Во время начальной записи (около 80 мс) доступ с магистрали к памяти запрещен.

В динамических БИС КР565РУ5 возможны случайные сбои интенсивностью до 10^{-3} 1/ч. Они вызваны в основном воздействием на кристалл α -частиц, искусственных микрооскопическими примесями радиоактивных элементов, содержащихся в корпусах микросхем [1], и сопровождаются инверсией заряда на запоминающих конденсаторах БИС.

При длительном хранении данных число ошибок в накопителе постепенно увеличивается; для их устранения необходимо периодическое контрольное считывание всех ячеек и перезапись информации после коррекции. Контрольное считывание в ОЗУ-СКМ реализовано аппаратно и начинается сразу после окончания начальной записи адресного кода (для повышения скорости выявления ошибок циклы



Структурная схема устройства ОЗУ-СКМ

обращения к накопителю следуют друг за другом без промешкутов [2]). Запрос на запись или выдачу данных в МПИ поступает в схему приоритета и ожидает окончания очередного цикла контрольного считывания. Таким образом, время выборки изменяется от одного до двух циклов работы.

Для обнаружения и коррекции ошибок используется код Хэмминга. При этом каждый из двух байтов хранимой информации защищен отдельными контрольными разрядами (их общее число равно 10), поэтому операция «запись байта» требует столько же времени, как «запись» и «считывание».

Борьба со случайными сбоями осуществляется следующим образом. При обнаружении одиночной или двойной ошибки в любом из байтов (в режиме контрольного считывания или чтения с магистралей) в один из четырех разрядов регистра состояний (РС) блока коррекции записывается Лог. 1 (в разряды РС (6), РС (14) при одиночной и РС (7), РС (15) при двойной ошибках). Затем она поступает в контроллер, который через БМУ выдает запрос на прерывание процессора. В случае корректируемого отказа в разряды РС (0)...РС (4), РС (8)...РС (12) записывается синдром ошибки соответствующего байта. Разряды РС (5), РС (13) определяют режим работы БК: Лог. 0 — коррекция байта включена, Лог. 1 — выключена. Одновременно с занесением информации в РС адрес ячейки с корректируемой ошибкой заносится в регистре адреса ошибки (РАО), расположенном в контроллере.

После получения разрешения от процессора в магистраль выдается вектор прерывания. Процессор читает регистры РС и РАО при соответствии вектора прерывания одиночной ошибке и локализует неисправность с точностью до запоминающего элемента. Затем данные считываются и вновь записываются в отказавшую ячейку (по адресу, считанному с РАО). Ошибка корректируемая, поэтому данные достоверны и бит информации восстанавливается.

С помощью системы прерываний ОЗУ-СКМ можно также вести статистический учет отказов накопителя (при повторении ошибок в одном и том же запоминающем элементе необходима его замена). Если такой учет не проводится, считывание содержимого РС необязательно.

По мере увеличения числа неисправных запоминающих элементов случайные сбои могут привести к возникновению некорректируемого отказа. Синдром и адрес ошибки не записываются в регистры РС и РАО, однако в разряды РС (7) или РС (15) заносится Лог. 1, инициирующая процедуру прерывания.

При подключении к МПИ платы ОЗУ-СКМ, работающей в режиме «горячего» резерва, накопитель защищается от ошибок более высокой кратности. Данные записываются по одним и тем же адресам основного и резервного устройств. Основная плата выдает ответный сигнал при условии окончания записи в резервной плате. В процессе чтения при отсутствии двойной ошибки данные и ответный сигнал поступают с основной платы, а при наличии двойной ошибки — с резервной, которая с этого момента становится основной. При возникновении в ней, в свою очередь, двойной ошибки производится обратное переключение. Таким образом, резервированная система из двух устройств ОЗУ-СКМ выдает искаженную информацию только в том случае, если ее невозможно скорректировать ни в основной, ни в резервной платах.

В резервном режиме можно изъять одну из плат для ремонта и вернуть на прежнее место без остановки работы системы. Появление одиночных и двойных ошибок, а также режим работы (основной-резервный) отображаются схемой индикации.

Вторая плата может быть применена не только для горячего резерва, но и для расширения объема памяти (информационная емкость составит 256 Кбайт).

Положение ОЗУ-СКМ в адресном пространстве МПИ (64 Кбайт) задается зонами по 8 Кбайт с помощью перемычек. Возможны два способа доступа к каждой из зон. При непосредственном доступе для выбора ячейки исполь-

зуются 16 адресных разрядов и обращение производится только к первому полублоку накопителя (17-й разряд выбора полублока принимается равным нулю). При страничном доступе местоположение ячейки внутри страницы (8 Кбайт) определяется 13-ю младшими магистральными разрядами, а номер одной на 16 страниц — содержимым регистра старших адресов (РСА), доступным по записи и чтению. Область адресного пространства магистралей, определяющая размер страницы, называется «окном». Его положение совпадает с одной из восьми зон и задается коммутацией перемычек. При этом необходимо, чтобы данная зона была открыта для доступа.

РСА имеет следующую структуру. Разряды РСА (0)...РСА (3) определяют номер страницы накопителя, РСА (5) указывает (при считывании), какое из двух ОЗУ-СКМ работает в данный момент в основном режиме. Запись в него Лог. 1 переключает основное устройство в резервный режим и наоборот. Разряд РСА (7) разрешает или запрещает запись информации в страницу, определяемому разрядами РСА (0)...РСА (3) (Лог. 0 — запись разрешена, Лог. 1 — запрещена). Попытка записать информацию в закрытую по записи страницу приводит к зависанию. Содержимое разрядов РСА (4) и РСА (6) равно Лог. 0 и Лог. 1 соответственно.

ОЗУ-СКМ целесообразно использовать в высоконадежных вычислительных системах, к быстродействию которых не предъявляется жестких требований.

Телефон 535-96-27, Москва

ЛИТЕРАТУРА

1. Бродский М. Уменьшение частоты случайных сбоев, вызываемых в динамических ЗУПВ действием альфа-частиц // Электроника.— 1980.— № 10.— С. 25—33.
2. А. С. 942137 СССР, МКИ G11 С 7/00. Устройство для управления регенерацией в блоках оперативной памяти / Д. А. Бруевич, Р. М. Воробьев, А. Г. Куликов.— Опубл. 1982, Бюл. № 25.

Статья поступила 10.08.88

РЕКЛАМА

Организация в короткий срок выполнит:

ремонт всех типов магнитных головок для НМД отечественного и зарубежного производства для ЭВМ серии ЕС и СМ;

ремонт вентиляторов и электродвигателей;
поставку ремней для дисководов типа зебра (эти работы заказчик оплачивает после удовлетворительных тестовых испытаний);

автоматизацию систем различного назначения на базе микро- и мини-ЭВМ с применением модулей КАМАК;
поставку средств программирования КАМАК (ассемблер, Фортран, Паскаль, KEYСИК, БЕЙСИК, Модула-2, Си, Форт).

Проведет комплексные исследования с использованием следующих методов анализа:

просвечивающая электронная микроскопия;
сканирующая электронная микроскопия;
микрорентгеноспектральный анализ;
электронная ОЖЕ-спектрометрия;
вторичная ионная МАСС-спектрометрия;
высокотемпературная оптическая микроскопия;
дифференциальный термический анализ;
рентгеноструктурный фазовый анализ;
широкий спектр химических исследований на современном импортном оборудовании.

Телефон 522-34-47, Москва

УДК 681.323

А. Михальски, В. Свицер

СЕМЕЙСТВО ГРАФИЧЕСКИХ УСТРОЙСТВ ВВОДА-ВЫВОДА

В семействе графопостроителей насчитывается ряд устройств, отличающихся форматом носителя, принципом конструкции, функциональными возможностями (табл. 1). Однако все они имеют привод с использованием шаговых двигателей, один и тот же управляющий МП Z80 (или аналог U880 производства ГДР), сходную структуру управляющего программно-обеспечения, стандартные аппаратные и программные интерфейсы. В состав семейства входят устройства ввода графической информации — два дигитайзера форматами А3 (DIGISS3) и А1 (DIGISS1). Оба работают по электромагнитному принципу с пассивной сеткой и имеют один и тот же блок управления с МП Z80. Используемый метод сравнения фазового сдвига сигналов, индицируемых в пассивных обмотках сеток измерительного стола, обеспечивает разрешающую способность 0,1 мм.

Графопостроитель MDG-1

Графопостроитель (рис. 1) предназначен для работы в системах на основе ПЭВМ (табл. 2). Его удобно использовать для подготовки простой конструкторской документации и технологических карт обработки деталей. Благодаря простоте обслуживания он очень хорошо подходит к школьным применениям в качестве учебного пособия и устройства, комплектующего рабочее место ученика и учителя. Встроенные генераторы специальных символов (кроме польского и русского алфавитов) и возможность работы с бумажной лентой длиной 2×6,5 м делают его удобным для применения в системах автоматизации научного эксперимента. Полный набор символов стандартного генератора представлен на рис. 2. По заказу графопостроитель может поставляться с другими специализированными символами.

Стандартные аппаратные и программный интерфейс обеспечивают удобное подключение к большинству современных компьютеров, работу с различными пакетами прикладных программ. По мере необходимости графопостроитель

Таблица 1

Основные параметры семейства графопостроителей

Основные параметры	MDG-1	GRAF85	MERA630	GRAFISS1	GRAFISS0
Формат	A4	A3	A3	A1	A0
Разрешающая способность, мм	0,2/0,1	0,1	0,1	0,1	0,05
Скорость, мм/с	80	200	230	350	600
Число пишущих элементов	4	6	4	8	8
Аппаратный интерфейс	CENTRONICS RS232C				
Программный интерфейс	DXY	HP-GL			
Конструктивное исполнение	Рулонный	Плоский	Рулонный	Рулонный	Рулонный

Примечание. Графопостроитель MERA630 — разработка Научно-производственного объединения систем управления в Катовицах.

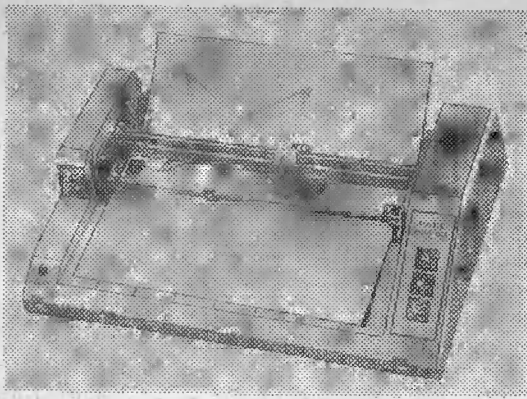


Рис. 1. Графопостроитель MDG-1

Таблица 2
Основные параметры графопостроителя MDG-1

Параметр	Значение
Формат	A4
Ширина бумаги, мм	100...216
Доступная программно длина бумаги, мм	6550
Разрешающая способность, мм	0,2 (0,1)
Число пишущих элементов	4
Скорость черчения, мм/с	80
Стандартные интерфейсы	CENTRONICS, RS232C, DXI
Графический язык	Графический, алфавитно-цифровой
Режимы работы	
Число графических команд	12
Напряжение питания, В	220±10; 15 %
Потребляемая мощность, ВА	20
Габаритные размеры, мм	305×195×90
Масса, кг	2

Примечание. По специальному заказу графопостроитель может поставляться в исполнении, предназначенном для бортовых применений, т. е. с питанием от источника постоянного напряжения 12 В.

тель поставляется с драйверами для распространенных прикладных программ, например для программы AUTOCAD фирмы AUTODESK, Inc. В графопостроителе используется метод перемещения бумажного носителя и пишущего элемента вдоль перпендикулярных осей X и Y. Для привода применены шаговые микродвигатели. У каждого микродвигателя ротор выполнен в виде гладкого постоянного магнита с электрически нанесенными пятью парами полюсов. Статор четырехсекционный, имеет десять геометрически выраженных полюсов. Управляющие сиг-

налы, поступающие на входы двигателей, представлены на рис. 3.

Зубчатая передача с устройством удаления люфта обеспечивает разрешающую способность (минимальный шаг) 0,2 мм для алгоритма управления $1/2$ и 0,1 мм для алгоритма $3/8$,

	SP	0	@	P	°	p		+ α	E	ю	Ю	П			
	32	48	64	80	96	112	128	144	160	176	192	208	224	240	
DC1	!	1	A	Q	a	q		Л	æ	Ł	а	я	А	Я	
17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	
DC2	"	2	B	R	b	r		т	β	Ɔ	р	Б	Р		
18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	
	#	3	C	S	c	s		†	š	š	ц	Ц	С		
	35	51	67	83	99	115	131	147	163	179	195	211	227	243	
	\$	4	D	T	d	t		†	š	š	г	м	Д	Т	
	36	52	68	84	100	116	132	148	164	180	196	212	228	244	
	%	5	E	U	e	u		—	š	ž	e	у	Е	У	
	37	53	69	85	101	117	133	149	165	181	197	213	229	245	
	&	6	F	V	f	v		—	ŋ	ž	φ	ж	Ф	Ж	
	38	54	70	86	102	118	134	150	166	182	198	214	230	246	
	'	7	G	W	g	w		—	λ	α	α	Г	В		
	39	55	71	87	103	119	135	151	167	183	199	215	231	247	
BS	08	0	H	X	h	x		—	ι	ε	с	х	ь	Х	Ь
	40	56	72	88	104	120	136	152	168	184	200	216	232	248	
)	9	I	Y	i	y		—	ι	ε	ц	ы	И	Ы	
	41	57	73	89	105	121	137	153	169	185	201	217	233	249	
LF	10	*	J	Z	j	z		—	ι	ε	ı	з	И	З	
	42	58	74	90	106	122	138	154	170	186	202	218	234	250	
LU	11	†	K	[k	{		—	ι	ε	к	ш	К	Ш	
	43	59	75	91	107	123	139	155	171	187	203	219	235	251	
	,	<	L	\	l	!		—	ι	ε	л	э	Л	Э	
	44	60	76	92	108	124	140	156	172	188	204	220	236	252	
CR	NC	-	=	M] m	}		—	ι	ε	м	ш	М	Ш	
13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
SO	14	.	>	N	^	n	~		—	ι	ε	н	ч	Н	Ч
	46	62	78	94	110	126	142	158	174	190	206	222	238	254	
SI	15	/	?	O	_	o	⊠		—	ι	ε	о	б	О	Б
	47	63	79	95	111	127	143	159	175	191	207	223	239	255	

Рис. 2. Набор символов встроенного генератора графопостроителя MDC-1

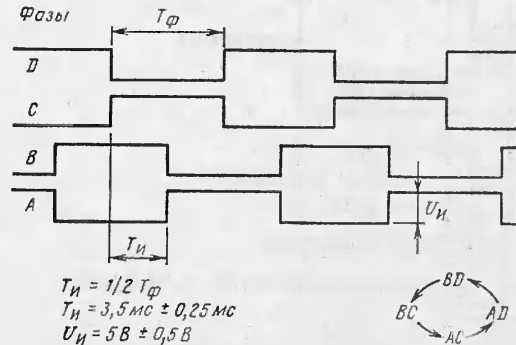


Рис. 3. Временная диаграмма сигналов управления микродвигателем (против часовой стрелки).

Для смены направления вращения изменяется последовательность фазовых сигналов (BD→AD→AC→BC→BD...)

в котором полный цикл управления состоит из последовательности восьми возбуждений. Для управления пишущими элементами (поднимания, опускания, смены цветов) используется такой же двигатель, как для привода бумаги

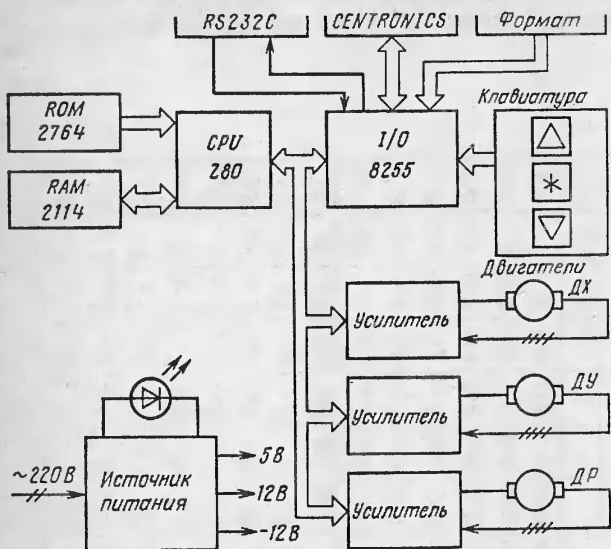


Рис. 4. Структурная схема графопостроителя MDG-1 с двумя интерфейсами: ДХ — перемещение пишущего элемента (каретки) вдоль бумаги; ДУ — перемещение бумаги; ДР — управление пишущим элементом: смена цвета и замена элемента; опускание; поднимание



Рис. 5. Клавиатура графопостроителя. Клавиша, обозначенная звездочкой, изменяет функции двух остальных

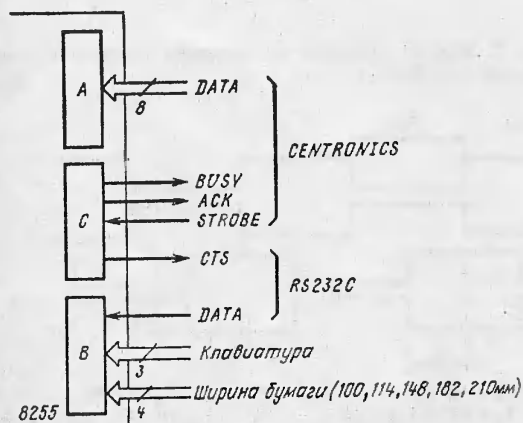


Рис. 6. Подключение внешних сигналов к портам ввода-вывода элемента 8255. Порт А работает в режиме 1, порт В — в режиме 0

или каретки. Все необходимые управляющие сигналы, вырабатываемые МП Z80, поступают на двигатели через усилители тока. Для сопряжения МП с остальными внешними устройствами (интерфейсными разъемами, датчиками ширины бумаги, пультом управления) предусмотрен один параллельный порт типа 8255 (рис. 4).

Управляющая программа и генератор символов хранятся в ЭРПЗУ типа 2764. В качестве ОЗУ используются два элемента типа 2114. Клавиатура из трех клавиш (рис. 5) обеспечивает прямой и обратный прогоны бумаги, смену цвета пишущего элемента, выполнение теста. При выполнении теста выводятся все символы встроенного генератора (см. рис. 2). Узлы графопостроителя питаются от источника постоянного напряжения 5 В, оснащенного индикатором на светодиоде. Для подключения источника питания не требуется сетевой розетки с заземляющим или зануляющим контактом. Способы использования отдельных портов модуля параллельного ввода-вывода типа 8255 показаны на рис. 6.

Работа с компьютером

Аппаратные интерфейсы. В графопостроителе могут быть одновременно задействованы два стандартных интерфейса — параллельный CENTRONICS (ИРПР-М) и последовательный RS232C (стык С2). Схема управления автоматически определяет, с помощью какого интерфейса графопостроитель подключен к компьютеру. Связь по параллельному интерфейсу осуществляется передачей 8- или 7-разрядных кодов согласно протоколу, представленному на рис. 7, по последовательному интерфейсу — 7-разрядного кода с одним битом останова и проверкой четности (рис. 8). Скорость обмена составляет 4800 Бод.

Графопостроитель поставляется также со специализированными интерфейсами, например параллельным ИРПР или для домашних компьютеров ATARI.

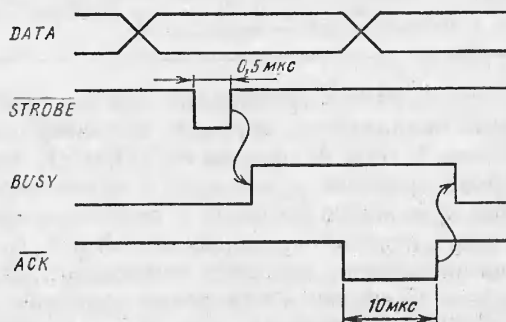


Рис. 7. Временная диаграмма сигналов параллельного интерфейса CENTRONICS. Все сигналы имеют ТТЛ-уровни

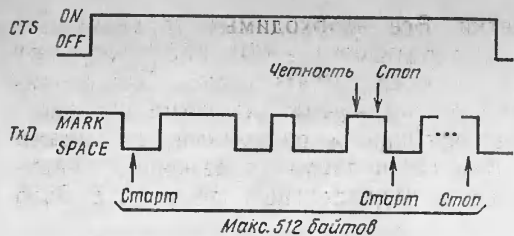


Рис. 8. Временная диаграмма сигналов последовательного интерфейса RS232C высокого (3...12 В) и низкого (-12...-3 В) уровней

Программный интерфейс. Для управления работой графопостроителя с компьютером используется стандартный графический язык DXU.

Мнемоника команды	Команды языка DXU Назначение
D	Черчение векторов абсолютное
F	Выбор начальной точки
H	Перемещение пишущего элемента в начальную точку
I	Черчение векторов относительное
J	Выбор цвета пишущего элемента
L	Выбор типа линии
M	Перемещение пишущего элемента абсолютно
P	Вывод символов
Q	Выбор направления вывода символов
R	Перемещение пишущего элемента относительно
S	Выбор размера символов
Z	Переход в текстовый режим

Режим работы. Графопостроитель имеет два основных режима работы — графический и тек-

УДК 681.06:332.1
Т. Синкевич

СИСТЕМА ОТЛАДКИ MSWP ДЛЯ 8-, 16- И 32-РАЗРЯДНЫХ МИКРОПРОЦЕССОРОВ

Основные этапы проектирования и отладки МПУ с помощью системы MSWP16/32 можно представить в виде схемы (см. рисунок). Вначале работы формулируются требования к аппаратной и программной частям МПУ, разрабатываются структура аппаратной части и алгоритмы программных модулей. В дальнейшем работа идет параллельно в аппаратной и программной ветвях с взаимным учетом получаемых результатов.

Создается детальная структурная схема программ, определяются и программируются на избранном исходном языке функции (задачи) программных модулей. Затем программные модули транслируются в результирующий код, соответствующий языку МП разрабатываемого МПУ. На следующем этапе модули отлаживаются и объединяются в одну программу, загружаемую в ПЗУ макета разрабатываемой аппаратной части. Правильность функционирования программы перед загрузкой в ПЗУ оценивается с помощью моделирующих программ.

В это же время конструкторы аппаратной части разрабатывают логическую и принципиальную схемы устройства, на основе которых строится макет, подлежащий первоначальной отладке. На очередном этапе макет аппаратной части и загруженная в ПЗУ результирующая программа проходят местную отладку через серию корректировок и проверок с помощью внутрисхемных эмуляторов, работающих в реальном масштабе времени.

После окончания отладочных работ проводятся функциональные испытания макета, анализируются результаты, строится прототип (опытный образец), подлежащий многосторонним эксплуатационным испытаниям. Положительный результат этих испытаний позволяет приступить к действиям, связанным с технической подготовкой производства. Ориентировочно трудоемкость основных отладочных работ для типового макета МПУ оценивается (в %) следующим образом: пуск и наладка макета — 18, отладка — 32, интеграция аппаратной и программной частей — 19,

После включения питания он находится в графическом режиме, предназначенном для вывода графической информации с помощью команд языка DXU. В текстовом режиме используются управляющие коды, назначение которых показано ниже. Текст составляется из символов, хранимых во внутреннем генераторе (см. рис. 2).

Десятичное значение	Мнемоника	Назначение
08	BS	Возврат каретки на один символ
10	LF	Новая линия
11	LU	Возврат к предыдущей линии
13	CR	Перевод каретки в начало линии
14	SO	Переход к дополнительному набору символов
15	SI	Переход к основному набору символов
17	DC1	Переход в текстовый режим
18	DC2	Переход в графический режим
29	NC	Смена цвета пишущего элемента
32	SP	Пробел

Описанный графопостроитель является относительно дешевым, но достаточно универсальным выходным графическим устройством, что обеспечивает ему довольно широкую область применений.

40-161, KATOWICE, ul. Armii Gzermonej, 101, Instytut Systemow Sterowania; tel. 585-671. 121099, Москва, ул. Чайковского, 11, Координационный центр Межправительственной комиссии по сотрудничеству социалистических стран в области вычислительной техники; тел. 255-43-66, Барток Ежи.

Статья поступила 7.03.89

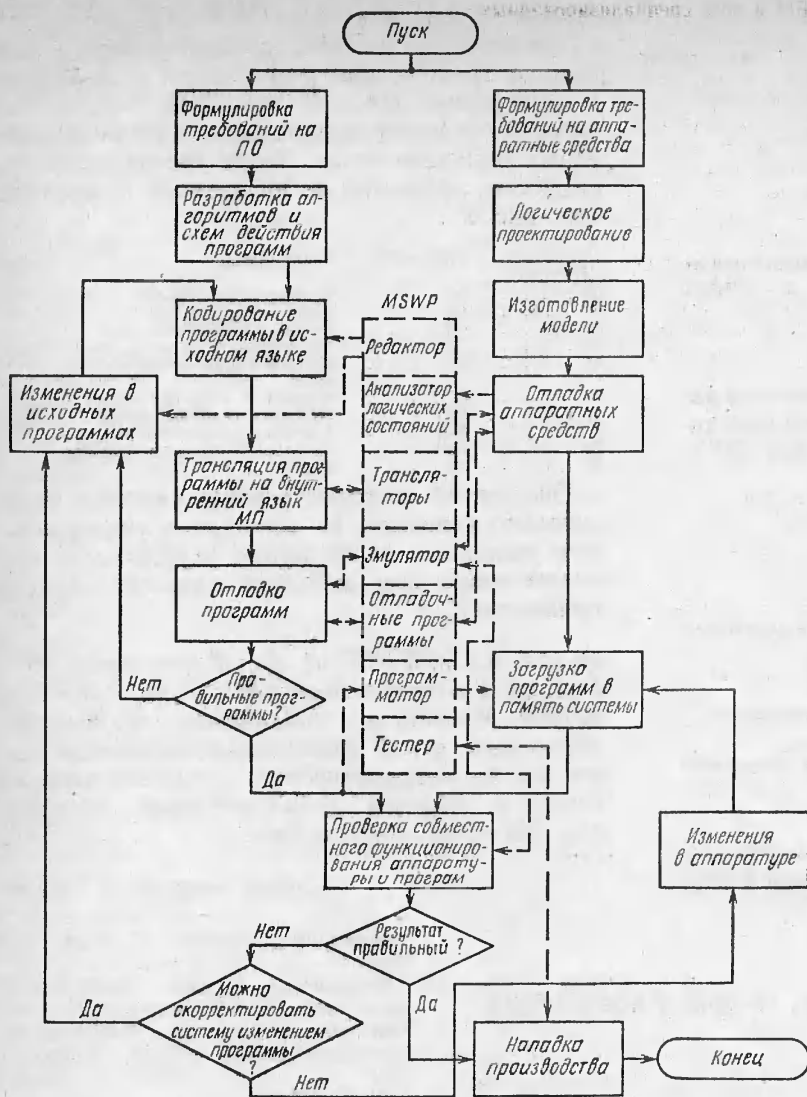
трансляция программ — 7, вывод листов — 20, разные работы — 4.

Отладочная система MSWP16/32 представляет собой многофункциональный набор аппаратных и программных инструментальных средств, построенных на базе ПЭВМ класса IBM PC/XT/AT.

Система имеет модульную структуру, что позволяет составлять требуемые конфигурации с использованием следующих аппаратных модулей: 16-разрядной ПЭВМ с внутренней памятью 640 Кбайт, дисплеем, жестким диском емкостью 20 или 40 байт, двумя накопителями ГМД 5.25", печатающим устройством Д-100 или Д-160.

В ПЭВМ должен быть встроен анализатор логических состояний на 24 канала с частотой синхронизации 75 МГц для 8 и 20 МГц для 24 каналов и памятью на 4096 состояний для 8 каналов.

В системе предусмотрены внутрисхемные эмуляторы для 8- и 16-разрядных микропроцессорных БИС 18080, 18085, Z80, 18035/18048/18049, 18031/18051, 18086/18088/180286. Эмуляторы выполнены в виде автономных модулей с собственными источниками питания, соединенных с центральным устройством посредством последовательного интерфейса ИРПС. Кроме того, они содержат свой эмуляционный процессор, заменяющий МП пользователя, а также



Этапы проектирования и отладки МПУ с использованием системы MSWP16/32

управляющие и запоминающие микросхемы, позволяющие вести полный контроль и управление работой налаживаемого устройства.

С помощью внутрисхемных эмуляторов выполняются следующие операции: прогон программ с максимальной скоростью МП при полном объеме памяти и числе входов-выходов; использование шести аппаратно реализуемых точек останова, причем некоторые из них могут строиться с учетом логических связей между параметрами останова;

полный контроль аппаратных управляющих сигналов; пользование 56-разрядной памятью трассировки, регистрирование 2048 шагов и анализ значения сигналов до и после точки останова; ввод и вывод программы «пользователь»;

просмотр и изменение содержимого памяти; потактовое выполнение программ; изменение содержимого регистров МП с переменным интервалом стробирования; использование встроенных программ ассемблера и дисассемблера; многократная посылка данных на произвольный выходной порт; диагностика памяти и блочные пересылки данных; загрузка и изменение содержимого памяти устройства пользователя.

В разработке находится внутрисхемный эмулятор для 32-разрядных БИС 180386.

В состав системы входят программы для следующих микросхем:

РПЗУ типа EPROM 12716...127512 и ПМЛ типа PAL с 20 и 24 выводами (плата, устанавливаемая в централь-

ном устройстве ПЭВМ с выносным пультом и колодками для программируемых БИС);

РПЗУ типа EPROM 12716...127512, 12816...12864 и ОЭВМ типа 18741...18752 (автономный модуль с собственным источником питания и последовательным интерфейсом ИРПС);

ПМЛ 40 типов микросхем PAL (автономный модуль с встроенным источником питания и последовательным интерфейсом ИРПС).

В состав аппаратных средств системы включены также: имитатор РПЗУ типа EPROM 12716...127512, легко приспосабливаемый для других ПЗУ (автономный модуль с управляемой буферной памятью 128 Кбайт, собственным источником питания и быстрым последовательным интерфейсом для связи с ПЭВМ, пригодным для отладки микроконтроллеров с управляющей программой, хранимой в ПЗУ); тестер для функциональной проверки, идентификации и анализа интегральных микросхем типов ТТЛ, МОП и КМОП с числом выводов, не превышающим 24, с собственным источником питания; ультрафиолетовая лампа для стирания микросхем РПЗУ типа EPROM.

Программное обеспечение системы MSWP16/32 в зависимости от реализуемых функций можно разделить на следующие группы:

- программы общего назначения;
- инструментальные пакеты программиста;
- управляющие программы аппаратных модулей;
- тестирующие программы.

Группу программ общего назначения составляет дисковая стандартная ОС PC-DOS с сопутствующим набором системных программ, позволяющая эффективно использовать все ресурсы ПЭВМ.

Инструментальные пакеты программиста дают возможность разработать и автономно отладить ПО проектируемого МПУ. Как правило, это кросспрограммы для различных МП, отличающихся от МП, примененного в ПЭВМ системы MSWP16/32.

В состав инструментальных пакетов входят следующие программы:

- редактор текстов;
- макросассемблеры;
- компиляторы языков ПЛ/М, Фортран, Паскаль, С;
- объединения результирующих модулей;
- преобразующие перемещаемые объектные файлы в абсолютные;
- библиотеки и программы управления библиотеками;
- отладочные;
- управления файлами (копирования, ликвидации, изменения атрибутов, изменения формата и т. п.);
- загрузки.

Инструментальные программы обеспечивают возможность модульного программирования с использованием различных языков, управляющие программы аппаратных модулей — удобную

эксплуатацию этих модулей. Эти программы в большинстве построены по принципу «дружеского отношения к пользователю» (user friendly), что значительно облегчает их изучение и эксплуатацию.

Управляющие программы аппаратных модулей, оснащенных собственным МП (все внутрисхемные эмуляторы) хранятся частично в РПЗУ этих модулей, что позволяет использовать их автономно (без ПЭВМ), но со значительным снижением числа реализуемых функций.

Тестирующие программы дают возможность проверить работоспособность

ПЭВМ и всех специализированных аппаратных средств системы с помощью встроенных диагностических процедур поиска неисправных функциональных модулей.

Система MSWP16/32 применяется в основном в автоматизации наиболее утомительных и трудоемких работ, выполняемых при проектировании, наладке и тестировании МПУ различного назначения. Кроме того, она может применяться для автоматизированного контроля элементов и узлов на различных этапах изготовления, программирования микросхем, отладки, тестирования и конечного контроля аппарату-

ры. Ее можно эффективно использовать в качестве комплексной учебной лаборатории для обучения разработчиков и программистов микропроцессорных систем.

121099, Москва, ул. Чайковского, 11, Координационный центр Межправительственной комиссии по сотрудничеству социалистических стран в области вычислительной техники; тел. 255-43-66; Барток Ежи

Статья поступила 11.04.89

РЕКЛАМА

СИСТЕМА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ

Состав системы:

персональный микрокомпьютер «Правец 16» или ИЗОТ 1037 с жестким диском и цветным монитором;
печатающее устройство;
модуль и программа для контроля и диагностики микропроцессорных устройств;
модуль и программа для записи в ПЗУ и ОЭВМ;
устройство для стирания ПЗУ и ОЭВМ;
редактор;
кроссассемблеры для 18080, 18086, 18046, 18051, 6502, 6800, 6805, 6809;

симуляторы;
дизассемблеры;
графопостроитель;
программа для вычерчивания электрических схем;
программа для разводки печатных плат.

Монтаж на месте. Обучение. Сервис.

Разработчик: Болгария, 7017, Русе, ул. Комсомольская, 8, ВТУ им. Ангела Кынчева, ОНИЛ по вычислительной технике. ТЕЛЕКС 62462 Телефон 5-07-34
Экспортёр: Болгария, 1113, София, ул. акад. Георги Бенчев, бл. 4, ВТП Баннимпекс. Телекс 23688 Телефон 70-30-06

УДК 681.3.06

М. В. Борденов

ОСОБЕННОСТИ ЯЗЫКА ПАСКАЛЬ ПЭВМ ЕС1840

Редактор языка Паскаль в ОС М86 оказался самым простым для изучения. Непрофессионалам он дает возможность применять ПЭВМ для ведения делового архива практически с первых минут, осваивать ОС М86 и различные ППП. Для этого круга пользователей предлагается ряд стартовых пакетных файлов, которые сразу погружают его в облюбованную среду.

Один из самых популярных пакетов составлен в полном соответствии с заводской документацией. Как видно из распечатки с экрана (рис. 1), после

формирования М-диска на него заносится БЕЙСИК или Паскаль с дисководом 2. Вся дальнейшая работа пользователя идет с М-дискон, что экономит ресурсы дисководов и дискет по износу.

Как правило, большинство работ не требует последующей записи на дискеты. Если она все же нужна, то организуется в конце сеанса работы системными средствами (команда КОП); также управляется и печать (команда ПЧ). Перед печатью системной командой НАП можно изменить принятые по умолчанию параметры печати.

Команда ЧТ введена только для того, чтобы весь пакетный файл можно было распечатать с экрана. Параметр в команде ДАТА инициирует появление подзаказки с форматом команды.

Для профессионалов, не владеющих языком, редактор в ОС М86 открывает простой путь для первого знакомства. Это облегчает вход в Turbo Pascal и MS DOS с их мощными средствами поддержки и многостажными меню

Краткая служебная программа, нашедшая широкое применение (рис. 2), оформлена как законченный модуль,

```

Program Dt;
uses Printer, Dos, Crt ;
VAR
  gggg,mm,dd,w : word ;
  hh,nn,ss,cs : word ;
BEGIN
  getdate (gggg,mm,dd,w) ;
  gettime (hh,nn,ss,cs) ;
  write(gggg,' ',mm,' ',dd) ;
  if (w-1) and (w<6) then
  case w of
    0:write(' воскресенье ');
    1:write(' понедельник ');
    2:write(' вторник ');
    3:write(' среда ');
    4:write(' четверг ');
    5:write(' пятница ');
    6:write(' суббота ');
  end ;
  writeln ('время ',hh,':',nn,':',ss,'.',cs);
END.

```

Рис. 2

используемый в других программах для вывода системного времени и текущей даты.

121002, Москва Г-2, Б. Власьевский, 7/24, кв. 12, Борденову М. В.; тел. 241-57-48

Сообщение поступило 24.02.88

```

КОНФИГУРАЦИЯ :
          НРЯД : 2
ПЕЧАТАЮЩЕЕ УСТРОЙСТВО : 1
АДАПТЕР СГНЦА С2 : 2
ПАМЯТЬ (КБ) : 256
М-ДИСК (КБ) : 256

```

```

1>Пакет 1:z.pak 1 2 и
1>ЧТ 1:z.pak
ЧТ $0
коп $3:=#2:basicM86.cmd
коп $3:=#2:sample.bas
$3:basicM86
коп $3:=#2:pascal.*
дата $0
1>КОП И:=2:basicM86.cmd

```

```

1>
      НОЯ 0      0:01:14  ПТ      1  АПР. 1988

```

Рис. 1

УДК 681.31.003.13:652.12

С. В. Казменко

НЕСКОЛЬКО ЭТЮДОВ РАЗРАБОТКИ ЗАДАЧ В СРЕДЕ ЭЛЕКТРОННЫХ ТАБЛИЦ

Таблица 1

С внедрением ПЭВМ в сферу организационного управления появилась возможность автоматизировать такие задачи, которые конечный пользователь обычно решает, не прибегая к услугам специалистов по обработке данных (программистов и разработчиков задач АСУ).

Широкий класс задач пользователь может самостоятельно решить с помощью электронных таблиц (ЭТ) (например, в среде пакетов Суперкалк [1], Варитаб* и др). Однако следует заметить, что каждый конкретный пакет поддержки ЭТ обладает определенными ограничениями.

Пакет Варитаб позволяет работать с таблицами, имеющими до 63 столбцов и 254 строк при общем объеме памяти на одну таблицу не выше 22 Кбайт. При этих ограничениях на основе одной или нескольких таблиц могут быть решены только небольшие задачи, в число которых не входят многие расчетные задачи организационного управления.

Выполнение заложенной в таблицу программы — это строго последовательное вычисление значений клеток ЭТ по одной из двух внутренних формул оператора ИФ (условие, формула 1, формула 2) в зависимости от выполнения или невыполнения внутреннего условия.

Этюд 1. Итоговые строки, содержащие выборку

Рассмотрим расчет общего числа поставок фруктов в магазины (табл. 1). Фрагмент табл. 1 с шириной колонок (Б и Ц), полностью отображающих формулы в строках 15, 16, 17, приведен в табл. 2.

Предположим, что все исходные данные (и формулы) в таблице сформированы. В клетку А2 последовательно вносятся значения 13, 17 и 18 (все номера магазинов, в которые отгружались фрукты). Результат автоматического пересчета записывается в соответствующую итоговую строку. После зане-

№пп	А	Б	Ц	Д	Е
1	Расчет поставок фруктов в магазины				
2					
3					
4	Номер магазина	Отгружено			
5		яблок	груш		
6	17	100	200	ИФ (А6=А2,Б6,0)	ИФ (А6=А2,Ц6,0)
7	18	100	200	ИФ (А7=А2,Б7,0)	ИФ (А7=А2,Ц7,0)
8	13	100	200	ИФ (А8=А2,Б8,0)	ИФ (А8=А2,Ц8,0)
9	17	200	200	ИФ (А9=А2,Б9,0)	ИФ (А9=А2,Ц9,0)
10	13	200	200	ИФ (А10=А2, Б10,0)	ИФ (А10=А2,Ц10,0)
11	17	300	100	ИФ (А11=А2, Б11,0)	ИФ (А11=А2,Ц11,0)
12	18	200	100	ИФ (А12=А2, Б12,0)	ИФ (А12=А2,Ц12,0)
13					
14	Итого				
15	13	ИФ...	ИФ...		
16	17	ИФ...	ИФ...		
17	18	ИФ...	ИФ...		

Примечание: в А2 задаем номер магазина

Таблица 2

№ пп	А	Б	Ц
14	Итого		
15	13	ИФ (А15=А2,СУМ (Д6:Д12),Б15)	ИФ (А15=А2,СУМ (Е6:Е12),Б15)
16	17	ИФ (А16=А2,СУМ (Д6:Д12),Б16)	ИФ (А16=А2,СУМ (Е6:Е12),Б16)
17	18	ИФ (А17=А2,СУМ (Д6:Д12),Б17)	ИФ (А17=А2,СУМ (Е6:Е12),Б17)

сения в клетку А2 последнего номера магазина (18) получим табл. 3.

Этюд 2. Выделение одного из нескольких итогов

При работе на 8-разрядных микроЭВМ часто возникает необходимость решения в среде ЭТ таких задач, которые информационно не входят в одну таблицу. В этом случае необходимо переместить данные из одной таблицы в другую. Целесообразно переносить ито-

ги, полученные в некоторой исходной таблице. Иногда из нескольких итогов исходной таблицы нужен только один, и разные итоги должны располагаться в одном и том же месте. Пример таблицы расчета итога за текущий год по операции (например, работе, теме НИОКР), которая выполняется несколько лет, приведен в табл. 4.

В строках с 11 по 99 находятся исходные данные, в клетках Б6, Ц6, Д6 — итоги за соответствующие (первый, второй, третий) годы выполнения опера-

* Пакет Варитаб, использующийся в научно-исследовательском институте атомных реакторов им. В. И. Ленина на ПЭВМ «Роботрон 1715», русифицирован специалистами ВЦ Димитровградского автоагрегатного завода. В настоящее время существует еще одна версия этого пакета — СПРИНТ [2]. Соответствие команд этих пакетов приведено в Приложении.

Таблица 3

№пп	А	Б	Ц	Д	Е
1	Расчет поставок фруктов в магазины				
2	18				
3					
4	Номер магазина	Отгружено			
5		яблок	груш		
6	17	100	200	0	0
7	18	100	200	100	200
8	13	100	200	0	0
9	17	200	200	0	0
10	13	200	200	0	0
11	17	300	100	0	0
12	18	200	100	200	100
13					
14	Итого				
15	13	300	400		
16	17	600	600		
17	18	300	300		

Примечание: в А2 задаем номер магазина

Таблица 4

№ пп	А	Б	Ц	Д
1	1989	1988	В1+1	С1+1
2	текущего	первого	Итого года	третьего
3			второго	
4				
5				
6			Б7+Ц7+Д7	
7	Номер этапа	первого	Стоимость года	третьего
8			второго	
9				
10				
11			1	
12	2		10 000	
13	3		5800	
14	4			12 000
15	5			9 000

Таблица 5

№пп	А	Б	Ц	Д
1	1989	1988	1989	1990
2	текущего	первого	Итого года	третьего
3			второго	
4				
5				
6			15 800	
7		0	15 800	0

ции, в клетке А1 — номер того текущего года, по которому необходимо перенести итоги в другую таблицу. В результате расчета в клетках Б7, Ц7, Д7 формируются либо ноль (если соответствующий год не текущий), либо итоговая сумма текущего года. Суммируя значения Б7, Ц7, Д7 в клетке А6, получаем итог текущего года. При первом проходе по строкам значение А6 вычисляется до того, как подсчитаны Б7, Ц7, Д7, поэтому необходим двукратный пересчет (табл. 5).

Операции подведения итогов за текущий год особенно полезны при использовании командных файлов (КФ).

Этюд 3. Использование КФ для организации расчета нескольких таблиц

Пусть имеется несколько исходных таблиц, аналогичных табл. 4, с различными первым годом выполнения операции. Требуется действия, которые для конкретного текущего года произведут операцию, рассмотренную в этюде 2. Для этого применяются две вспомогательные таблицы — таблица-задание (ТаЗ — табл. 6) и таблица-заготовка КФ. Год, для которого в каждой исходной таблице необходимо выделить итог, задается в клетке А2 ТаЗ, имена этих таблиц — в клетках столбца А, начиная с А5. Значение, заносимое в А4, будет рассмотрено позже. Для работы с таблицами, отмеченными в ТаЗ, необходимы следующие действия:

загрузить таблицу-заготовку КФ на чистое рабочее поле и список исходных таблиц, сформированный в ТаЗ — в ее столбцы (Б и Д);

Таблица 6

№пп	А
1	Текущий год
2	1989
3	Первая свободная строка
4	10
5	ИМЯ 1
6	ИМЯ 2
7	ИМЯ 3
8	ИМЯ 4
9	ИМЯ 5

очистить лишние строки таблицы-заготовки (сформированный таким образом на рабочем поле КФ показан в табл. 7);

отпечатать сформированную заготовку в виде КФ;

запустить полученный КФ.

Функции каждой строки сформированного КФ одной исходной таблицы (пример 6-й строки, обрабатывающей таблицу с именем ИМЯ1):

очистка рабочего поля (/НД);

загрузка очередной исходной таблицы (/Т ИМЯ1, В — команда загрузки)

Таблица 7

№ пп	А	Б	Ц	Д	Е
1					
2					
3					
4					
5					
6	/НД/Т	ИМЯ1	,В/ТТа3,4А2,А1,3 !!/С	ИМЯ1	,ПВ
7	/НД/Т	ИМЯ2	,В/ТТа3,4А2,А1,3 !!/С	ИМЯ2	,ПВ
8	/НД/Т	ИМЯ3	,В/ТТа3,4А2,А1,3 !!/С	ИМЯ3	,ПВ
9	/НД/Т	ИМЯ4	,В/ТТа3,4А2,А1,3 !!/С	ИМЯ4	,ПВ
10	/НД/Т	ИМЯ5	,В/ТТа3,4А2,А1,3 !!/С	ИМЯ5	,ПВ

в рабочую область; загружается в ся (таблица);

Загрузка из ТаЗ номера текущего года (/ТТаЗ, ЧА2, А1,3 — в ячейку А1 рабочей области из ТаЗ загружается только часть этой таблицы, состоящая из значения ячейки А2, в которой указан год);

двукратный пересчет таблицы (!!); сохранение пересчитанной таблицы (/С ИМЯ1, ПВ; — сохраняется все содержимое рабочей области и перезаписывается файл).

Все рассмотренные выше действия по формированию и запуску на исполнение КФ также можно оформить в виде КФ. Единственное неопределенное значение — строки-заготовки, которые нужно чистить. В этом случае надо чистить все, начиная со строки, указанной в клетке А4 ТаЗ.

Достоинства рассмотренного этюда: позволяет реализовать пакетный режим для действий, которые нецелесообразно и даже вредно выполнять в диалоге, задавая команды вручную.

Этюд 4. Получение итогов в задаче с переменным числом исходных таблиц

При сложении итогов исходных таблиц, приведенных в этюде 2, в таблицу, содержащую заготовку КФ, нужно добавить столбец названий исходных таблиц. После этого формируется и исполняется КФ, показанный в табл. 8. В его заготовке — максимальное число командных строк переноса данных из одной исходной таблицы (такие заго-

товки команд могут быть в строках с 5-й по 99-ю, при этом число исходных таблиц — 95). Лишние строки, для которых не хватило исходной таблицы, должны быть очищены после переноса в заготовку КФ списка имен.

Этюд 5. Реализация технологии автоматического расчета задачи с переменным числом исходных таблиц

Примеры, рассмотренные в этюдах 2, 3, 4, могут быть выполнены автоматически. Для этого необходимо: произвести пересчет, аналогичный рассмотренному в этюде 2, для некоторого переменного числа исходных таблиц;

сформировать из частных результатов исходных таблиц итоговую таблицу, получив ее в виде документа;

применить параметр «первая свободная строка» (табл. 6, клетка А4). Дополнительный параметр нужен для автоматического построения КФ из заготовок, с его помощью можно стереть все лишние строки, начиная с «первой свободной».

Технология автоматического решения комплексной задачи может быть реализована следующей цепочкой КФ.

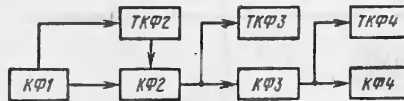


Таблица 8

№ пп	А	Б	Ц	Комментарий
1	/Н	Д		Очистка рабочего поля
2	/Т	РЕЗ,В		Загрузка итоговой таблицы
3				
4	/Т	ИМЯ1	,ЧА6,А1,3	Загрузка итога: из первой исходной таблицы в клетку А1
5	/Т	ИМЯ2	,ЧА6А2,3	из второй — в А2
6	/Т	ИМЯ3	,ЧА6,А3,3	из третьей — в А3
7	/Т	ИМЯ4	,ЧА6,А4,3	из четвертой — в А4
8	/Т	ИМЯ5	,ЧА6,А5,3	из пятой — в А5
...
100	!!			Пересчет окончательной итоговой таблицы

Каждый последующий КФ подготавливается предыдущим по заготовкам КФ2, КФ3 и КФ4, находящимся соответственно в таблицах-заготовках командных файлов (ТКФ): ТКФ2, ТКФ3 и ТКФ4.

КФ1 формирует КФ2, КФ2—КФ3 (единственная функция), КФ3—КФ4, а также выполняет действия, рассмотренные в этюде 3 для каждой исходной таблицы, КФ4 осуществляет сборку и пересчет (действия этюда 4).

Обоснуем необходимость именно четырех (а не меньшего числа) последовательно работающих КФ. КФ3 и КФ4 последовательно обрабатывают некоторое переменное (Х—5) число исходных таблиц, где Х — заданный в ТаЗ номер первой свободной строки. В заготовках ТКФ3 и ТКФ4 находятся команды обработки исходных таблиц, причем ТКФ3 и ТКФ4 рассчитаны на их максимально допустимое число. В процессе формирования КФ (КФ3 или КФ4) лишние строки стираются, начиная с первой не занятой, которая указана в ТаЗ, и кончая последней, где присутствуют команды обработки. Сокращенный вариант КФ2 имеет следующий вид.

/НД
/ТКФ3, В

Очистка рабочего поля
Загрузка на рабочем поле заготовки ТКФ3

/ТТаЗ, ЧА4:А99, И5

Перенос списка имен из ТаЗ в формируемый КФ3

/БА 10:Н99

Очистка строк (с 10-й указано в ТаЗ — по 99-ю)

/ОДВСЕ, —
ДКФ3, ИСП, П

Оформление текущей сформированной таблицы в виде исполняемого файла КФ3

/ИКФ3

Запуск КФ3

В КФ2 присутствует заданный в ТаЗ параметр 10, но КФ — просто текстовый файл, содержащий команды (последовательность нажатия на соответствующие клавиши, которую должен выполнить пользователь, работая вручную). Таким образом, КФ1 должен быть разделен с КФ2 для получения в КФ2 сформированной команды, содержащей параметр 10.

Данный этюд демонстрирует возможность организации пакетного режима вычислений на основе данных, хранящихся в электронных таблицах. Организация пакетного режима необходима при большом объеме вычислений без пользователя.

Этюд 6. Использование шифратора

Для расчета затрат на командировки требуется составить табл. 9.

Информационные строки заполняются значениями только первых двух столбцов (А и Б), так как столбец Ц может быть сформирован заранее (или в процессе выполнения КФ), а значения столбцов Д, Е, Ф переносятся в таблицу

№пп	А	Б	Ц	Д	Е	Ф						
1 Расчет затрат на командировки												
2	Шифр города	Число дней	Сумма	Город	Стоимость проезда	Суточные						
3												
4												
5							4	10	E5+Ф5	Ялта	100	B5×3,50
6							5	10	E6+Ф6	Сочи	120	B6×3,50
7							8	5	E7+Ф7	Минск	60	B7×3,50
8							6	7	E8+Ф8	Баку	140	B8×3,50
9							7	2	E9+Ф9	Тула	40	B9×3,50
10							3	7	E10+Ф10	Киев	80	B10×3,50

1. Дойл У. Табличный процессор Суперкалк для персонального компьютера: Пер. с англ. — М.: Финансы и статистика, 1987.
2. Система табличной обработки данных в электронных таблицах. СПРИНТ. Описание применения. — Таллинн: Эст. НИО ВТИ, 1987.

Статья поступила 24.02.89

Таблица 11

расчета командировок из шифратора города (табл. 10).

Таблица 10

№ пп	Д	Е	Ф
1 Шифратор командировок			
2	Москва	20	Ц2×3,50
3	Киев	80	Ц3×3,50
4	Ялта	100	Ц4×3,50
5	Сочи	120	Ц5×3,50
6	Баку	140	Ц6×3,50
7	Тула	40	Ц7×3,50
8	Минск	60	Ц8×3,50

№ пп	А	Б	Ц	Комментарий
1	/ND			Очистка рабочего поля
2	=Ц5			Установка в Ц5
3	E5+Ф5			Занесение в Ц5 формулы
4	/PЦ5,Ц6:Ц99			Размножение формулы
5	/ТГорода, ЧД	4	.Д5	Перенос данных из шифратора
6	/ТГорода, ЧД	5	.Д6	
7	/ТГорода, ЧД	8	.Д7	
8	/ТГорода, ЧД	6	.Д8	
9	/ТГорода, ЧД	7	.Д9	
10	/ТГорода, ЧД	3	.Д10	
11	/ТГорода, ЧД	200	.Д11	
12	/ТГорода, ЧД	200	.Д12	

Последовательность автоматического расчета таблицы:

загрузка бланка таблицы расчета командировок в чистое рабочее поле (шапка — первые четыре строки табл. 9);

занесение исходных данных в столбцы А и Б;

сохранение частично заполненной таблицы с именем КОМА;

загрузка заготовки КФ и столбца с шифрами городов в нее;

оформление исполняемого КФ и его запуск.

Действия по формированию таблицы с использованием шифратора достаточно трудоемки, однако их часть (все, что выполняется после ввода исходных данных) может быть выполнена в виде КФ.

В КФ (табл. 11) можно не стирать лишние строки, которым не хватило соответствующих строк таблицы КОМА, они будут переносить данные из пустой строки шифратора (200-й).

ПРИЛОЖЕНИЕ

Соответствие команд пакетов Варитаб и СПРИНТ

Аббревиатура команды		Функции
Варитаб	СПРИНТ	
А — архив	С(S) — сохранить	Перенос таблицы или ее части из рабочей области на магнитный носитель в виде файла
Т — таблица	З(Z) — загрузить	Перенос таблицы или ее части с магнитного носителя в рабочую область
Н(N) — новая таблица	О — очистка	Полная очистка рабочего поля
Б(B) — бланк	Г(G) — гасить	Очистка части рабочего поля
А — альтернатива	Р(R) — режим	Установка режимов работы
Г(G) — гарантия	Б(B) — блокировать	Установка запрета на изменение содержимого группы клеток таблицы
У(U) — убрать гарантию	Н(N) — нет блокировки	Снятие запрета на изменение содержимого клеток
П(P) — править	И(I) — изменить	Перенос содержимого некоторой клетки в строку ввода для его изменения с последующим помещением в любую другую клетку

В(W) — вставить	В(W) — вставить	Вставка в таблицу на рабочем поле новой строки или нового столбца
Л(L) — ликвидировать	У(U) — убрать	Удаление строки, столбца или файла (существующей таблицы)
М — менять	С(S) — сдвиг	Перенос строки или столбца на новое место
К — копия	Д(D) — дубль	Копирование блока клеток
Р(R) — размножить	Т — тираж	Размножение содержимого одной или нескольких клеток из линейного диапазона (часть строки или столбца). Содержимое каждой тиражируемой клетки переносится в одну или несколько клеток
Ф(F) — формат	Ф(F) — формат	Задание формата строки, столбца, всей таблицы или отдельного элемента
З(Z) — заголовки	Ш(С) — шапка	Установка-снятие верхнего и (или) левого заголовка
Д(D) — два окна	Э — экран	Установка-снятие двухоконного режима работы
О — отпечатать	П(P) — печать	Печать части (или всей) таблицы, находящейся в рабочей области
И(I) — исполнить	М — макро	Исполнение командного файла
Я(Q) — QUIT	К — конец	Выход из среды электронных таблиц в среду ОС

В. Н. Степанов

ПРОСТЕЙШИЙ МУЛЬТИЗАДАЧНЫЙ ПЛАНИРОВЩИК

В системах управления на основе МП-комплекта КР580 иногда требуется выполнять параллельно несколько программ. Если в системе отсутствует возможность мультизадачности, то можно использовать исполнительную программу-планировщик. Хотя планировщик и не обладает такими же возможностями по мультизадачности, как полная ОС, но он очень прост для понимания и использования*.

Работа планировщика заключается в следующем. Каждая прикладная программа в определенных точках выполнения освобождает процессор и вызывает планировщик, который сохраняет состояние данной программы и инициирует выполнение следующей программы из фиксированной последовательности. Преимущество такой схемы: система не квантуется по времени и не прерывает манипуляций с данными в памяти, поэтому не требуется явной защиты переменных от конфликтов доступа.

Один из путей, приводящих к более полной мультизадачности ПО, — перемещение обращения к планировщику из программ в подпрограммы (например, подпрограммы ввода-вывода), которые часто вызываются. Для несинхронизированных программ используются только вызовы в точках логического останова программы, таких как «конец основного цикла». Для увеличения разрешающей способности процессорного времени следует увеличивать число вызовов планировщика в программе.

Основное обращение к планировщику — CALL PROG_STOP. С помощью планировщика можно создать некоторые простейшие ОС. Например, чтобы отложить выполнение программы до момента, когда будет установлен семафор, можно применить способ

```
REPEAT TASK_STOP UNTIL
SEMAPHORE:=1;
SEMAPHORE:=0;
```

Аналогично, чтобы начать выполнение задачи после установки семафора, можно использовать конструкцию

```
SEMAPHORE:=1;
TASK_STOP;
```

При параллельном выполнении двух независимых программ следует вста-

В. С. Горбачев, В. Л. Попов

КРАТКОЕ СООБЩЕНИЕ

КРОСС-СИСТЕМА АВТОНОМНОГО ПРОГРАММИРОВАНИЯ ДЛЯ РОБОТА РМ-01 С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ ТРЕХМЕРНОЙ МАШИННОЙ ГРАФИКИ

Кросс-система автономного программирования робота РМ-01 (КАПР) предназначена для автономной разработки, создания и отладки управляющих программ для робототехнических комплексов (РТК) с промышленными роботами РМ-01. КАПР позволяет:

автономно определять и корректировать точки позиционирования ПР, эффективно обнаруживать и исправлять грамматические, алгоритмические и геометрические ошибки в управляющей программе;

тестировать программы с использованием средств трехмерной машинной графики, наблюдая движение манипулятора на графическом дисплее;

переносить автономно отлаженную управляющую программу в устройство управления СФЕРА-36 с линиями связи или гибким магнитным диском;

решать задачи компоновки РТК и размещения оборудования (создавать модели оборудования, перемещать их в пространстве, менять точку обзора, масштаб).

Аппаратные средства системы базируются на ЭВМ СМ4, дооснащенной графическим растровым дисплеем Электроника МС7401.

Программное обеспечение состоит из следующих компонентов:

отладчика для языка программирования ARPS, предоставляющего пользователю развитые средства контроля правильности выполнения программы;

подсистемы графического трехмерного моделирования, используемой без автономного программирования;

модуля определения и редактирования точек.

Предлагаемую систему можно поставить на ПЭВМ типа ЕС1841, «Искра 1030.11», «Электроника 85».

Телефон 289-51-75, Москва; адрес 220093, г. Минск, ул. Притыцкого, 65, НПО «Гранат»

* Г. Григорьев В. Л. Программное обеспечение микропроцессорных систем. — М.: Энергоатомиздат, 1983. — 207 с.

2. Howard M. Multitasking scheduler works, without OS // EDN. — Vol. 27, N 15. — P. 194—195.

```

МПК КР580
*****
МУЛЬТИЗАДАЧНЫЙ ПЛАНИРОВЩИК
*****
РАЗМЕЩЕНИЕ ДАННЫХ
*****
** СТЕК И НАЧАЛЬНЫЙ АДРЕС **
*****
***** К ПРОГРАММЕ 1 *****
DS 128
STACK1: DS 8
PROG1 INIT: DS 2
*****
***** К ПРОГРАММЕ 2 *****
DS 128
STACK2: DS 8
PROG2 INIT: DS 2
*****
УКАЗАТЕЛЬ АКТИВНОЙ ПРОГРАММЫ
PROG: DS 1
*****
УКАЗАТЕЛИ СТЕКА *****
PROG1 SP: DS 2
PROG2 SP: DS 2
*****
*** ОТСЮДА ЗАПУСКАЕТСЯ ***
***** ПЛАНИРОВЩИК *****
SCHEDULER: LXI H, PROG1
SHLD PROG1 INIT
LXI H, STACK1
SHLD PROG1 SP
LXI H, PROG2
SHLD PROG2 INIT
LXI H, STACK2
SHLD PROG2 SP
MVI A, 01
STA PROG
JMP PROG START
*****
*** ОТСЮДА ПЛАНИРОВЩИК ***
*** ВЫЗВАЕТСЯ ПРИКЛАДНЫМИ ***
***** ПРОГРАММАМИ *****
PROG STOP: PUSH PSW
PUSH B
PUSH D
PUSH H
LDA PROG
CPI 01
JZ PROG1 STOP
CPI 02
JZ PROG2 STOP
JMP ERROR1
*****
*ПЕРЕХОД НА ОБРАБОТЧИК ОШИБОК*
*****

```

```

*****
PROG1 STOP: LXI H, 0000H
DAD SP
SHLD PROG1 SP
JMP PROG START
*****
PROG2 STOP: LXI H, 0000H
DAD SP
SHLD PROG2 SP
JMP PROG START
*****
PROG START: LDA PROG
CPI 01
JZ PROG2 START
CPI 02
JZ PROG1 START
JMP ERROR2
*****
*ПЕРЕХОД НА ОБРАБОТЧИК ОШИБОК*
*****
PROG1 START: MVI A, 01
STA PROG
LHLD PROG1 SP
SPHL
POP H
POP D
POP B
POP PSW
RET
*****
PROG2 START: MVI A, 02
STA PROG
LHLD PROG2 SP
SPHL
POP H
POP D
POP B
POP PSW
RET
*****

```

```

*****
ПРИМЕР ПРИКЛАДНЫХ ПРОГРАММ
*****
РАЗМЕЩЕНИЕ ДАННЫХ
*****
СЧЕТЧИКИ КОЛИЧЕСТВА ВЫПОЛНЕНИЙ
***** ПРОГРАММ И СЕМАФОР *****
PROG1 COUNT: DS 2
PROG2 COUNT: DS 2
SEMAPHORE: DS 1
*****
* ПРОГРАММА 1 ВЫПОЛНЯЕТСЯ ПОСЛЕ *
* КАЖДЫХ 16 ВЫПОЛНЕНИЙ ПРОГРАММЫ 2 *
PROG1 :LHLD PROG1 COUNT
INX H
SHLD PROG1 COUNT
LDA SEMAPHORE
CPI 10H
JZ T2
CALL PROG STOP
JMP T1
T2 :XRA A
STA SEMAPHORE
JMP PROG1
*****
PROG2 :LHLD PROG2 COUNT
INX H
SHLD PROG2 COUNT
LDA SEMAPHORE
INR A
STA SEMAPHORE
CALL PROG STOP
JMP PROG2
*****

```

Рис. 1. Исполнительная программа-планировщик

Рис. 2. Пример обращения к планировщику

вить обращение к PROG_STOP в конце каждого основного цикла:

```

PROCEDURE TASK1;
BEGIN
  WHILE TRUE DO BEGIN
    TASK1 CODE HERE
    TASK_STOP;
  END;
END.

PROCEDURE TASK2;
BEGIN
  WHILE TRUE DO BEGIN
    TASK2 CODE HERE
    TASK_STOP;
  END;
END.

```

Телефон 9-98-66, Таганрог

Сообщение поступило 9.09.88

УДК 681.3.06

С. В. Пашенков, М. Ф. Стерин

МОБИЛЬНАЯ ФАЙЛОВАЯ СИСТЕМА С ПРОГРАММНЫМ ИНТЕРФЕЙСОМ, СОВМЕСТИМЫМ С UNIX

Программное обеспечение (ПО) специализированных микропроцессорных (МП) устройств, предназначенных для применений в АРМ, узлах приема-передачи информации и т. п., как правило, не содержит сколько-нибудь развитых средств разработки программ (например, ПО интеллектуального терминала ТАР-34 производства ВНР).

Отсутствие средств разработки программ можно компенсировать кросс-системой программирования, подобной [1]. Она позволяет переводить программы, написанные на Small-C [2] — достаточно широком подмножестве языка Си [3], в коды МП К580ИК80, готовые для непосредственного выполнения. Кроме того, можно программировать на ассемблере, связывать отдельно транслируемые модули в единую программу и отлаживать в диалоге на мини-ЭВМ с помощью интерпретатора кодов МП.

Кросс-система программирования не компенсирует отсутствия поддержки времени выполнения, в частности файлового ввода-вывода. Нами создано ПО, поддерживающее файловую систему на ГМД. Зависимость от конкретной аппаратуры минимальна.

Требования к файловой системе

Язык Си и операционная система UNIX [4] тесно связаны: среда выполнения программ в UNIX стала практически стандартной для языка Си. Совместимость программного интерфейса нашей файловой системы и файловой системы UNIX необходима для обеспечения переносимости программ из UNIX на микроЭВМ.

Для обеспечения такой совместимости нужно иметь возможность использовать место, которое занимал удаленный файл сразу, без уплотнения диска. Поэтому требуется, чтобы был общий для всего диска список свободных блоков и блоки файла могли быть разбросаны по диску в произвольном порядке.

Надежность ГМД не очень высока, и часто возникают сбойные блоки. Файловая система должна опреде-

лять и обходить сбойные блоки автоматически, без вмешательства пользователя.

Чтобы достичь переносимости, надо исключить обращения к НГМД напрямую. Вместо этого для взаимодействия файловой системы и НГМД используется специальная программа-драйвер, представляющая ГМД как массив блоков с прямой адресацией. Доступ к блокам предоставляется по номерам (от 0 до максимального значения, зависящего от типа НГМД).

Реализация файловой системы

Для упрощения программирования и (что более важно) уменьшения объема памяти под программы управления решено было отказаться от многоуровневого каталога. На диске всегда есть один каталог, и в нем перечислены все файлы этого диска. Такое решение позволило зафиксировать область диска, содержащую каталог, что значительно упростило программу распечатки каталога (она может быть размещена в ПЗУ).

Все пространство диска разделяется на три части: каталог, таблица распределения блоков (bat) и пространство данных.

Каталог имеет по одному входу для каждого файла. Вход в каталог содержит информацию о файле: имя файла, дату последней модификации (в символьном виде), код защиты (один символ, г или w), длину файла в байтах (символьное представление десятичного числа), физический номер первого блока файла. Чтобы получить физические номера остальных блоков файла, используется таблица распределения блоков.

Таблица распределения блоков (bat) — это массив с количеством элементов, равным числу блоков на диске. Каждый элемент bat описывает состояние соответствующего блока. Если блок распределен для некоторого файла, соответствующий элемент bat содержит либо физический номер следующего блока в этом файле (если такой есть), либо код того, что этот блок последний в файле. Блоки, содержащие bat и каталог, помечены особо. Свободные блоки, а также обнаруженные при инициализации плохие блоки, также помечены соответствующим кодом в bat (см. рисунок). Таблица распределения бло-

ков записывается на диск несколько раз, чтобы избежать катастрофической потери информации при ошибке ввода-вывода в области таблицы.

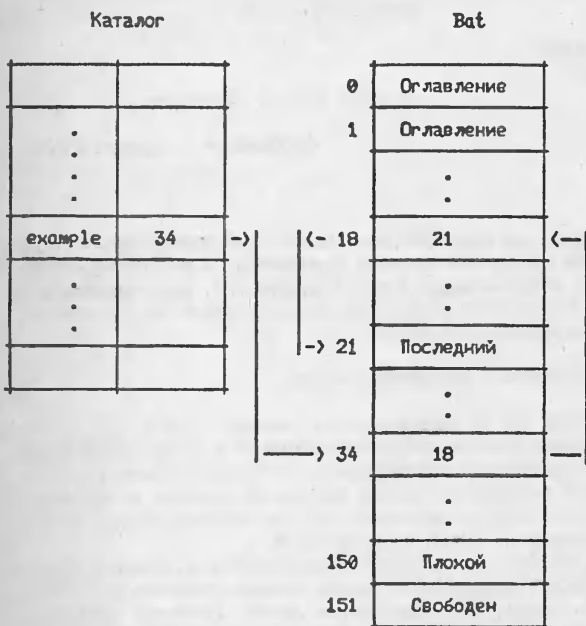
Все остальное место на диске занимают непосредственно блоки файлов. Размеры и число блоков на диске, расположение и размеры областей, максимальное число файлов зависят от конкретного типа устройства и драйвера. Эти параметры, заданные с помощью '#define', можно легко изменить.

Файлы в файловой системе различаются по именам, состоящим из обязательного номера диска, двосточия и символьной строки (до 14 байт). В имя файла могут входить любые символы, кроме пробела.

Файловая система рассматривает файлы как последовательность из нуля или более байтов. Вся сложная структура накладывается на файлы программами пользовательского уровня. Это упрощает реализацию и использование примитивов файловой системы и позволяет программам иметь файлы любой желаемой структуры.

Для каждого открытого файла есть буфер размером в блок, через который проходит весь обмен с диском. В буфере всегда содержится тот блок файла, на который установлен текущий указатель чтения-записи. Запросы пользователя на обмен с файлом приводят к чтению или записи буфера, и только при пересечении границ блока происходит реальный обмен с диском. Если буфер не был модифицирован, он не записывается. Файл может быть открыт несколько раз. При этом не возникнет никаких неприятностей, если не пытаться модифицировать один и тот же блок с помощью разных дескрипторов файла одновременно. Открытый файл можно удалить. При этом запись о нем в каталоге пропадет, а информационные блоки останутся занятыми и с ними можно работать вплоть до последнего закрытия файла, при котором блоки станут свободными. Это хороший способ создания временного файла без имени.

Описанная файловая система реализована на ТАР-34 с НГМД MF-3200 для дисков EC5274 одинарной плотности с 77 дорожками по 26 128-байтных секторов. Размер блока файловой системы равен 1644 байта (13 секторов). Число блоков на диске — 152 (76 дорожек, 77-я дорожка зарезервирована для подмены сбойных секторов). Максимальное число файлов на диске — 38. Таблица распределения блоков расположена в секторах 1—2 (основная) и 3—4 (резервная) дорожки 0; каталог, занимающий 19 секторов, — в секторах 8—26 дорожки 0. Такое расположение каталога диктуется программой распечатки каталога из ПЗУ ТАР-34. Объем программ управления файловой системой — 1400 строк на Small-C, или 12 Кбайт кода МП.



Распределение блоков для файла example.

Файл состоит из трех блоков: 34, 18, 21. (В файловой системе всего 152 блока)

Описание программного интерфейса

Файловая система реализуется с помощью набора функций, вызываемых из программы пользователя и подключаемых на этапе редактирования связей из библиотеки. Параметры файловой системы определены в файле, который можно включить в программу пользователя оператором '#include'. Перед началом работы с диском он должен быть смонтирован, а по завершении работы с ним — размонтирован.

Функция mount(dev); делает ГМД доступным для работы с ним. Она возвращает 0, если все в порядке, и отрицательное значение — в случае каких-либо ошибок (монтирование уже смонтированного диска, ошибка ввода-вывода, неверное значение dev и т. п.).

Функция umount(dev); завершает работы с указанным ГМД, если на нем нет открытых файлов. В случае успеха возвращается значение 0, а в случае ошибки — отрицательное значение (неверное значение dev, dev не смонтирован, на dev есть открытые файлы, ошибка ввода-вывода и т. п.).

Чтобы писать или читать файл на ГМД, его необ-

ходимо сначала явно открыть двумя функциями `open` или `creat`:

```
int fd;
fd=open(name, mode);
```

Аргумент `name` — это символьная строка, содержащая имя файла. `Mode` равно нулю для чтения, единице — для записи с начала файла (при этом старое содержимое теряется) и двум — для чтения и записи (модификация). Попытка открыть несуществующий файл — ошибка. Функция

```
fd=creat(name, pmode);
```

создает файл `name` (если его нет) или ускакает его до нулевой длины. `Pmode` определяет его код защиты. 0400 (`pmode` <0577 означает разрешение только читать, 0600 (`pmode` <0777 — читать и писать. Другие значения недопустимы. При какой-либо ошибке возвращается значение `fd<0`, в противном случае — дескриптор файла.

Весь ввод-вывод осуществляется функциями:

```
int n_read, n_written;
n_read=read(fd, buf, n);
n_written=write(fd, buf, n);
```

Первый аргумент обеих функций — дескриптор файла, второй — буфер в пользовательской программе, а третий — число подлежащих пересылке байтов. Возвращается количество фактически переданных байтов. При чтении оно может оказаться меньше, чем запрошенное число. Возвращенное значение 0 означает конец файла, а отрицательное сигнализирует об ошибке. Несовпадение запрошенного и фактически переданного количества байтов при записи также сигнализирует об ошибке.

По завершении работы с файлом его следует закрыть. Это делается функцией `close` (`fd`); `fd` — дескриптор открытого файла. Возвращается «0», если все в порядке, и отрицательное значение в случае ошибки.

Для удаления файла служит функция `unlink` (`name`); `name` — символьная строка, содержащая имя файла. Соответствующий вход оглавления диска очищается. Если указанный файл не открыт, то его блоки освобождаются. В противном случае это делается при последнем закрытии файла.

При работе с файлами нормальный ввод-вывод — последовательный, но при необходимости файл можно читать и записывать в произвольном порядке. Для этого служит функция

```
int offset, origin;
seek(fd, offset, origine);
```

В файле с дескриптором `fd` текущая позиция передвигается на позицию `offset`, которая отсчитывается от места, задаваемого аргументом `origin`. `Origin` может иметь значения от 0 до 5. Значения 0, 1, 2 указывают на то, что значение `offset` отсчитывается от начала файла, текущей позиции в файле, конца файла соответственно. Значение `offset` в этом случае интерпретируется как количество байтов. Если `origin` имеет значения 3, 4, 5, то значение `offset` отсчитывается так же, как и для 0, 1, 2 (но `offset` интерпретируется как количество блоков файловой системы). При этом для `origin=3` указатель устанавливается на начало соответствующего блока, для `origin=4` смещение в блоке остается неизменным, а для `origin=5` — устанавливается на конец блока либо на конец файла (для последнего блока). Функция `seek` возвращает новое положение указателя в тех же единицах, в которых было запрошено смещение, либо отрицательное значение в случае ошибки.

Для переименования файла служит нестандартная функция

```
rename(oldname, newname);
```

`oldname` — символьная строка со старым именем файла,

`newname` — символьная строка с новым именем файла без лидирующего номера диска и двоеточия. Эта функция просто корректирует соответствующий вход оглавления. Возвращается «0», если все в порядке, и отрицательное значение в случае ошибки.

Имеется дополнительная функция `inifs` (`dev`), которая строит пустую файловую систему на диске `dev`. Содержимое диска теряется. Возвращается «0», если все в порядке, и отрицательное значение в случае ошибки (диск смонтирован, ошибка ввода-вывода и т. п.). При инициализации каждый блок диска проверяется на работоспособность, поэтому эта функция выполняется довольно долго.

Выводы

Программы, работающие с диском с помощью описанной файловой системы, можно полностью отладить, пользуясь средствами разработки программ, предоставляемыми инструментальной ОС UNIX или ей аналогичной [5]. В частности, для TAP-34 написаны программы копирования, слияния, удаления и переименования файла, простейший экраный редактор текстов и др. Ни одна из этих программ не потребовала изменений в работе с файлами после отладки их на ЭВМ СМ4 под управлением ИНМОС. Эксплуатация файловой системы показала ее удобство, гибкость и надежность.

Телефоны 462-34-85, 344-51-79, Москва

ЛИТЕРАТУРА

1. Стерин М. Ф. Кросс-система для разработки программ в кодах микропроцессоров K580ИК80, INTEL 8080/8085. — В кн.: Вопросы проектирования и диагностики вычислительных устройств и комплексов. — М.: ИИЭУМ, 1986. — С. 77—81.
2. James E. H. The Small-C Handbook. — Prentice-Hall, Reston, Virginia, 2209, 1984. — 225 p.
3. Керниган Б., Ричи Д. Язык программирования Си. — М.: Финансы и статистика, 1985. — 280 с.
4. Банахан М., Раттер Э. Введение в операционную систему UNIX. — М.: Радио и связь, 1986. — 343 с.
5. Беляков М. И., Ливеровский А. Ю., Семик В. П., Шяудкулис В. И. Инструментальная мобильная операционная система ИНМОС. — М.: Финансы и статистика, 1985. — 232 с.

Статья поступила 15.03.88

РЕКЛАМА

Организация предлагает микропрограммные контроллеры для вычислительных комплексов, которые позволяют значительно расширить возможности Ваших ЭВМ.

Для микроЭВМ типа «Электроника-60» (ДВК, ВУМС, ИЦ-80, МС1201, МС1201.02, МС1202):

контроллер накопителей на магнитной ленте СМ5300.ИЗОТ 5003;

устройство управления магнитными дисками ЕС5061, СМ5400;

параллельный интерфейс И2М;

4-канальное устройство последовательного обмена; электронный диск 1 Мбайт.

Для мини-ЭВМ СМ1420, МС4 устройство управления магнитными дисками ЕС5061.

Для ППЭВМ ЕС1840, ЕС1841, «Искра 1030»:

контроллер накопителей на магнитной ленте СМ5300;

устройство подготовки данных ЕС9004;

телеграфный модем.

Принимаем заказы на разработку различных контроллеров и программно управляемых устройств.

Наш адрес: 220107, г. Минск, а/я 283. Тел. 46-48-21, 45-01-81.

УДК 681.3.066

Г. А. Мастерова, Т. А. Полякова

ОРГАНИЗАЦИЯ ДИАЛОГА И ДОКУМЕНТИРОВАНИЯ В СИСТЕМАХ АВТОМАТИЗАЦИИ НА ОСНОВЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Разработанное специальное программное обеспечение (ПО) организует диалог при вводе исходных данных и документировании результатов обработки без традиционного программирования (например, при использовании Фортрана — без применения операторов форматного ввода-вывода). Запросы для диалога и таблицы для документирования оформляются в виде страниц текстовой информации (в первом случае — текст запросов и символы вводимых данных их функциональной группы, размещаемые в поле экрана; во втором — таблицы на стандартном листе формата 11). Текстовая информация запросов подготавливается в автономном режиме с помощью любой программы ввода символов с клавиатуры дисплея (например, «Редактор»), записывается на какой-либо носитель. На этапе компоновки ПО системы из загрузочных модулей эта информация вводится в буферы, предусмотренные в соответствующих программных модулях.

В модуле, реализующем диалог, выполняется отображение запросов страницы, номер которой задает оператор (рис. 1). Вводимые значения (после перевода в вещественный формат, если это числовые значения) записываются в двоичный буфер, отведенный для данной страницы. Последовательность обеспечивается выполнением двух подпрограмм (номер страницы используется в качестве параметра в обращении). Предусмотрены режимы корректировки введенных данных, регистрации страниц на бланке печатающего устройства и на магнитной ленте, считывание страниц с магнитной ленты, регистрация каталога.

В модуле, реализующем документирование, выполняется отображение таблиц заданной страницы или регистрация их на бланке печатающего устройства с помощью единственного обращения к соответствующей подпрограмме (рис. 2). При регистрации поле страницы очерчивается рамкой и снабжается штампом. Данные выбираются из двоичного буфера, соответствующего заданной странице.

ПО оформлено в виде двух модулей, составленных на языке Фортран IV, и подпрограмм (Макроассемблер) с общим объемом, примерно, 2 Кслов (подпрограммы для диалога — 1 Кслов). Оно может быть использовано в составе ПО любой системы автоматизации. Специфика последней определяет размеры и число текстовых и двоичных буферов, которые можно легко откорректировать, и формы текстовых страниц. Связь с ОС присутствует лишь в двух фрагментах, обеспечивающих ввод и вывод символов. Из библиотеки стандартных под-

```

ДАТА           : 04.12.87      /
ВРЕМЯ          : 09-15-00     /
С ***** ПАСПОРТНЫЕ ХАРАКТЕРИСТИКИ
ЯРКОСТЬ СВЕЧЕНИЯ, ЛМ/ММ      : 16      /
КОЭФФИЦИЕНТ КонтРАСТА        : 6       /
РАЗМЕР РАВОЧЕГО ПОЛЯ, ММ     : 160 , 210 /
СКОРОСТЬ ПОСТРОЕНИЯ ЛИНИИ, М/С : 100   /
ВРЕМЯ СТИРАНИЯ, С            : 0.5     /
ВРЕМЯ ЗАПОМИНАНИЯ, МИН      : 15     /
ТОЛЩИНА ЛИНИИ, ММ           : 0.8     x
    
```

Рис. 1. Страница исходных данных

ПАСПОРТНЫЕ ХАРАКТЕРИСТИКИ				ТАБЛИЦА
№ П/Л	НАИМЕНОВАНИЕ	ОБОЗНАЧЕНИЕ	РАЗМЕРНОСТЬ	ЗНАЧЕНИЕ ФИЗИЧЕСКОЙ ВЕЛИЧИНЫ
1	ЯРКОСТЬ СВЕЧЕНИЯ	P1	ЛМ/ММ	16,0
2	КОЭФФИЦИЕНТ КонтРАСТА	P2		6,0
3	РАЗМЕР РАВ. ПОЛЯ	IR	ММ	160,0
				210,0
4	СКОР. ПОСТРОЕНИЯ ЛИНИИ	SP	М/С	100,0
5	ВРЕМЯ СТИРАНИЯ	SS	С	0,5
6	ВРЕМЯ ЗАПОМИНАНИЯ	UZ	МИН	15,0
7	ТОЛЩИНА ЛИНИИ	VL	ММ	0,8

Рис. 2. Страница результатов

программ используются подпрограммы перевода данных из символьного формата в числовой и обратно.

Разработанное ПО позволяет свести к минимуму затраты на реализацию диалога и документирования, так как требуется лишь подготовить формы страниц, откорректировать размеры буферов и выполнить компоновку модулей без их предварительной доработки. При большом объеме вводимых данных (более 40 запросов) объем памяти, занимаемый ПО, существенно меньше, чем при традиционном способе организации диалога. В последнем случае для программирования одной страницы запросов и ввода параметров требуется примерно 500 16-рядных ячеек памяти. Объем памяти, занимаемой модулем ввода вне зависимости от количества страниц, составляет примерно 1,5 Кячеек.

Последняя версия разработана для ОС RT-11. Модули могут выполняться, будучи записанными в перепрограммируемое запоминающее устройство, перезапись их в оперативную память не требуется.

Телефон 291-99-38, Ленинград

Сообщение поступило 29.12.87

УДК 681.3.066

И. Е. Байдан, Н. Г. А. Гильен, М. М. Гаджиев, Д. С. Г. Депрес

КОМПЛЕКС СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ НА МИКРОЭВМ

Внедрение МП-систем для решения системотехнических задач требует повсеместного применения так называемых систем разработки прикладных программ электронных устройств на машинных языках конкретных микропроцессоров (МП) [1, 3]. Существующие операционные системы (ОС) в большинстве случаев слабо обеспечивают пользователя в системотехнических задачах необходимым и эффективным инструментом создания электронной аппаратуры на МП-системах. Действительно, микроЭВМ, предназначенные для выполнения вычислительных процессов, не требуют в большинстве случаев предельного использования ресурсов МП, а также не вынуж-

дают считаться с особенностями его работы в реальном масштабе времени. Однако когда возникает вопрос применения МП-системы в предельных своих возможностях (а в системотехнических разработках это наиболее частый случай), реализованный алгоритм, показавший себя вполне приемлемым при эмуляции в кросс-системах, может оказаться неэффективным, а чаще и непригодным в реальных условиях.

На последней стадии испытаний, когда проверяются не только программы, но и схемные решения и, что важнее всего, их объективное сочетание в реальном масштабе времени, разработчик чаще всего оказывается беспомощным, так как программная часть электронного устройства кросс-системы не имеет внешнего, достаточно мощного сервисного программного обеспечения (ПО) и существует вне ОС, посредством которой создана. В этом случае мониторы реального времени, как единственные средства запуска и отладки в прикладной МПС программной части электронного устройства, оказываются весьма слабыми, и к их достоинствам следует отнести лишь простоту вместе с малым объемом занимаемой памяти в системе.

Предлагается система, в которой сочетаются достоинства программ мониторов прикладных МП-систем с возможностями ОС, ориентированных для разработки электронных устройств. При этом основная проблема — противоречивое сочетание широких возможностей ОС с малым объемом занимаемой памяти в адресном пространстве МП. Решение задачи удалось найти в способе обработки пользовательских файлов: манипулирование последними происходит в адресах их физического расположения. Таким образом, разработана ОС физических файлов (ОСФФ).

В предлагаемой ОСФФ опущены традиционные для других ОС предварительные назначения имен переменных и меток ветвления. Пользовательские файлы заносятся и обрабатываются в физических адресах прикладной программы. Это позволяет реализовать все процедуры создания и отладки прикладных программ в виде директив (команд ОСФФ), при этом сразу отпадает требование наличия в составе системы разработки больших объемов ОЗУ, например для системы МОС-2 — 48К.

Объем памяти, занимаемый ОСФФ в адресном пространстве МП, составляет 4 Кбайт, а область ОЗУ, требуемого для работы системы, не превышает 128 байт. Таким образом, ОСФФ на период разработки и отладки прикладных программ (или для создания новых на уже действующих МП-системах) может быть включена в состав МП-систем, так как фактически не теснит адресное

пространство пользователя. В ОСФФ имеются все общепринятые директивы мониторов [2—4], комплекс программных средств, расширяющих оперативность и возможности этих директив, а также подпрограммы, действующие в качестве посредников между пользователем и ресурсами системы. В ОСФФ программно реализованы возможности работы в мнемонических кодах языка ассемблера МП К580ВМ80 и организован набор средств перемещения файлов с обработкой единиц для сохранения программного смысла как внутри перемещаемого файла, так и при обращениях к нему извне.

Предлагаемая система (рис. 1) содержит информационный комплекс подготовки данных «Аккорд-05—06» и микроЭВМ К-1—20 («Электроника МС2702»). Комплекс «Аккорд-05—06» (состоящий из видеотерминала ВТА 2000-10 и кассетного накопителя КНМЛ СМ5211) предназначен для диалоговой работы оператора с системой и загрузки в оперативную память К-1—20 отлаживаемых прикладных программ. Кроме того, в систему могут входить печатающее устройство (ПУ), присоединяемое к последовательному каналу обмена информацией через порт К580ВВ51, и дополнительное ОЗУ требуемой емкости, включаемое в системную шину К-1—20. В распечатках листингов (рис. 2) соответственно приведены подпрограммы обмена информацией между К-1—20 и видеотерминалом ВТА 2000-10. Аппаратная распайка кабеля связи для указанного обмена приведена на рис. 3.

Рассмотрим систему директив ОСФФ, которую можно условно разделить на следующие группы команд: работы с памятью микроЭВМ, регистрами МП, текстовыми файлами; загрузки; редактирования программ; подготовки и отладки прикладных программ; формирования внешних файлов. Часть команд после их запуска завершается автоматически, далее управление микроЭВМ переходит к так называемому основному циклу ОСФФ, при котором в начале рабочей строки на экране дисплея высвечивается символ «!». Другая часть команд переходит к выполнению только после нажатия клавиши <BK> — «возврат каретки».

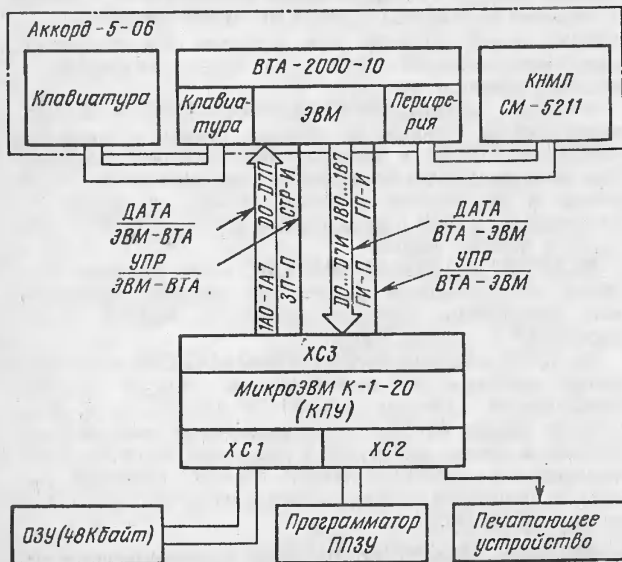


Рис. 1. Комплекс средств для схематических разработок

710C:	F5	PUSH	PSW ;
710D:	F5	PUSH	PSW ;
710E:	DBF6	IN	F6 ; Вывод
710B:	E640	ANI	40 ; данных
7102:	CADE71	JZ	710E ; из микроЭВМ
7105:	F1	POP	PSW ;
7106:	D3F4	OUT	F4 ;
7108:	3E0F	MVI	A, 0F ;
710A:	C36F73	JMP	736F ;

а

7355:	C5	PUSH	B ;
7356:	B603	MVI	B, 03 ;
7358:	0EC9	MUI	C, 09 ; Задержка
735A:	00	DCR	C ;
735B:	C25A73	JNZ	735A ; ≈ 10мс
735E:	05	DCR	B ;
735F:	C25873	JNZ	735E ;
7362:	C1	POP	B ;
7363:	C9	RET	;
7364:	CD7773	CALL	7377 ;
7367:	CA6473	JZ	7364 ;
736A:	E67F	ANI	7F ;
736C:	F5	PUSH	PSW ;
736D:	3E0A	MVI	A, 0A ; Ввод
736F:	D3F6	OUT	F6 ; данных
7371:	3E0B	MVI	A, 0B ; в микроЭВМ
7373:	D3F6	OUT	F6 ;
7375:	F1	POP	PSW ;
7376:	C9	RET	;
7377:	CD5573	CALL	7355 ;
737A:	DBF6	IN	F6 ;
737C:	E620	ANI	20 ;
737E:	C8	RZ	;
737F:	DBF5	IN	F5 ;
7381:	C9	RET	;

б

Рис. 2. Драйверы параллельного обмена информацией между К-1—20 и ВТА 2000-10 (а) и наоборот (б)

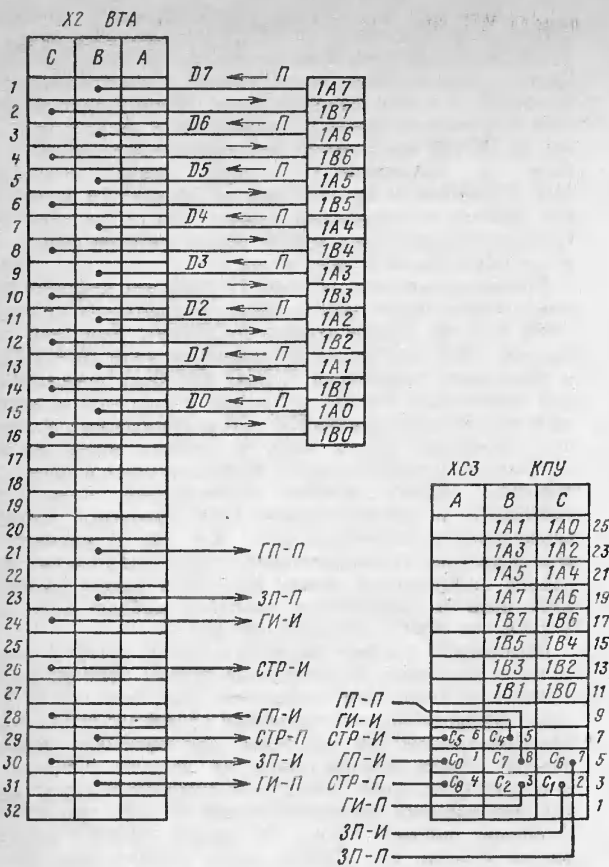


Рис. 3. Аппаратная распейка кабеля связи

Команды работы с памятью

DADDR1/ADDR2 — вывод содержимого области памяти с адресами ADDR1 (начальный) и ADDR2 (конечный) на экран дисплея в 16-ричных числах и формате (дамп памяти).

DTADDR1/ADDR2 — то же что и для команды DADDR1/ADDR2, но информация, выводимая на экран дисплея, дублируется распечаткой на телетайпе.

FADDR1/ADDR2XX — заполнение всех ячеек памяти в области ОЗУ с адресами ADDR1 и ADDR2 числом (константой) XX.

CADDR1/ADDR2/ADDR3 — побайтно сравнение двух областей памяти M1 и M2. M1 начинается с адреса ADDR1 и заканчивается адресом ADDR2, M2 начинается с адреса ADDR3.

MADDR1/ADDR2/ADDR3 <BK> — копирование массива данных побайтно, начинающегося с адреса ADDR1 и заканчивающегося адресом ADDR2, в область ОЗУ, начинающегося с адреса ADDR3.

SADDR — открытие файла в области ОЗУ с адреса ADDR для занесения (редактирования) данных побайтно в каждую отдельную ячейку. Чтобы оставить неизменным содержимое открытой ячейки, нажимается клавиша «пробел», после чего на экране ВТА индицируется адрес следующей ячейки и ее содержимое. Для выхода из команды необходимо нажать клавишу «возврат каретки». В области ПЗУ этой командой можно только индицировать каждую ячейку памяти.

STADDR — то же, что и в команде SADDR, но с копированием всего процесса работы с файлом данных на телетайп.

Команды работы с регистрами МП

XX — индикация всех регистров сверхоперативной памяти МП, а также счетчика команд и указателя стека.

XR — индикация и редактирование каждого отдельного регистра МП (R=A, B, C, D, E, H, L).

XF — редактирование содержимого регистра признаков (флагов) МП (F=S, Z, AC, CY, P). После нажатия клавиши F выбирается клавиша первой буквы обозначения редактируемого признака.

Команды загрузки данных

R <ПД> — занесение информации из буферного ОЗУ экрана видеотерминала в ОЗУ микроЭВМ. Первые (находящиеся рядом) два байта воспринимаются как начальный адрес области памяти для занесения данных побайтно. Знаки, не принадлежащие 16-ричной системе счисления, игнорируются.

Команды работы с текстовыми файлами

N <ПД> — вывод текстовой информации, содержащейся на экране ВТА-2000-10, для распечатки на телетайпе.

WADDR <ПД> — занесение текстовой информации с экрана ВТА в область ОЗУ, начиная с адреса ADDR.

QTADDR1/ADDR2 — вывод текста из области памяти, ограниченной адресами ADDR1 и ADDR2, на экран дисплея с одновременной распечаткой на телетайпе.

QADDR1/ADDR2 — то же, что и для QTADDR1/ADDR2, но без распечатки текста.

Команды редактирования программ на ассемблере МП К580ИК80

BADDR1/ADDR2 — вывод (индикация) программы на ассемблере из области памяти ADDR1 и ADDR2 на экран дисплея ВТА в мнемонических и машинных кодах с физическими адресами переходов.

BTADDR1/ADDR2 — то же, что и для BADDR1/ADDR2, но с копированием выводимой программы распечаткой на телетайпе.

AADDR — открытие файла с адреса ADDR и занесение (редактирование) мнемонических символов операций и операндов (т. е. ввод программы) в область ОЗУ, начинающегося с физического адреса ADDR. При этом на экране дисплея индицируется адрес ячейки памяти и ее содержимое в символах команд МП К580ВМ80. Чтобы оставить содержимое данной команды или операнда без изменения, нажимается клавиша «пробел»; для выхода из команды — клавиша «возврат каретки».

MADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — копирование таблиц данных из области памяти с адресами ADDR1 и ADDR2 в область ОЗУ (с адреса ADDR3). При этом происходит исправление операндов команд обращений к упомянутым таблицам данных в программе, находящейся в ОЗУ с физическими адресами ADDR4 (начало) и ADDR5 (конец).

MLADDR1/ADDR2/ADDR3 <BK> — копирование программ на ассемблере из области памяти, ограниченной физическими адресами ADDR1 и ADDR2, в область ОЗУ (с адреса ADDR3).

MLADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — копирование программ на ассемблере из области памяти, ограниченной адресами ADDR1 и ADDR2, в область ОЗУ (с адреса ADDR3) с одновременным исправлением операндов команд обращений в программе, находящейся в физических адресах ОЗУ ADDR4 (начало) и ADDR5 (конец), к программе, скопированной в области памяти ОЗУ (с адреса ADDR3).

MMADDR1/ADDR2/ADDR3 <BK> — восстановление работоспособности (исправление адресов внутренних переходов) программы, находящейся в области ОЗУ, начинаю-

щейся с адреса ADDR1 и ADDR2 и являющейся работоспособной в области памяти с начальным адресом ADDR3.

MMADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — то же, что и для команды MMADDR1/ADDR2/ADDR3/⟨BK⟩, однако одновременно производятся исправления адресов операндов команд обращения в программе, находящейся в области ОЗУ (ADDR4 и ADDR5), к восстановленной программе, начальный адрес которой ADDR1.

MKADDR1/ADDR2/ADDR3 — исправление адресов внутренних обращений к подпрограмме и переходов в программе, находящейся в области ОЗУ M1 и предназначенной для работы в другой области памяти M2. M1 ограничена адресами ADDR1 и ADDR2. M2 начинается с адреса ADDR3.

ZPADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — вставка программы с адресами ADDR1 и ADDR2 в программу области ОЗУ с адресами ADDR4 и ADDR5. Точка вставки ADDR3.

ZTADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — вставка таблиц, находящихся в памяти с адресами ADDR1 и ADDR2, в программу области ОЗУ с адресами ADDR4 и ADDR5. Точка вставки ADDR3.

KADDR1/ADDR2 — удаление пустых команд (NOP) в программе области ОЗУ с адресами ADDR1 и ADDR2 («сжатие программы удалением пустых команд»).

YPADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — исправление операндов команд обращений в программе области ОЗУ с адресами ADDR4 и ADDR5 к программе, которая физически переместилась из области памяти с адресами ADDR1 и ADDR2 в область памяти с начальным адресом ADDR3.

YTADDR1/ADDR2/ADDR3/ADDR4/ADDR5 — исправление операндов команд обращений в программе ОЗУ с адресами ADDR4 и ADDR5 к таблицам, переместившимся из области с адресами ADDR1 и ADDR2 в область, начинающуюся с адреса ADDR3.

Команды подготовки и отладки прикладных программ

GADDR⟨BK⟩ — пуск программы с заданного адреса ADDR.

GADDR1, ADDR2⟨BK⟩ — пуск программы с заданного адреса ADDR1 и остановка по адресу ADDR2 (с передачей управления по достижению ADDR2 ОСФФ).

GADDR1, ADDR2, ADDR3 — пуск программы с заданного адреса ADDR1 и остановка по первому (одному из двух) встречающемуся (ADDR2 или ADDR3) адресу.

G, ADDR ⟨BK⟩ — пуск программы с начального адреса, находящегося в счетчике команд МП, и остановка по адресу ADDR.

G, ADDR1, ADDR2 — пуск программы с начального адреса, находящегося в счетчике команд МП, с остановкой по первому (одному из двух) встречающемуся (ADDR1 или ADDR2) адресу.

G⟨BK⟩ — пуск программы с адреса, находящегося в счетчике команд МП.

GADDRH — трассировка. Для трассировки (покомандного режима выполнения программ, находящихся в области памяти ОЗУ) используются два символа клавиатуры терминала: H и P. По символу H исполняется только одна команда, адрес которой находится в счетчике команд МП. После исполнения одной команды управление передается основному циклу ОСФФ. По символу P в реальном масштабе времени (т. е. не по отдельным командам) выполняются подпрограммы в трассируемой программе, после чего снова может производиться покомандное выполнение программы по символу H.

По директиве GADDRH используется первая команда программы, начинающейся с адреса ADDR, и дальнейший процесс выполнения останавливается. На экране дисплея индицируется код операций выполненной команды, ее адрес и мнемоническое наименование. Кроме того, индицируется состояние всех регистров сверхоперативной

памяти МП, состояние счетчика команд и указателя стека.

Одновременно разработчику предлагается следующая команда трассируемой программы, которая также может быть выполнена при однократном нажатии клавиши H. Однако перед выполнением каждого следующего шага программы (последовательным нажатием клавиши H) система ОСФФ находится в основном цикле и, следовательно, возможно редактирование как состояния регистров и флагов МП, так и самих команд или операндов исполняемой в режиме трассировки программы.

Если перед командой вызова подпрограммы все же нажать клавишу H, то произойдет вход в подпрограмму при условии, что последняя находится в области памяти ОЗУ. Однако если произошел вход в подпрограмму, то выйти из нее можно и в реальном масштабе времени. Для этого нажимается клавиша P. Операционная система при этом задает оператору вопрос: «Действительно ли Вы желаете выйти из подпрограммы?» (TO THE NEXT RET?). При повторном нажатии клавиши P исполняется подпрограмма (до команды RET включительно) в реальном масштабе времени. На экране дисплея оператору будет предложена для очередного исполнения первая команда после выхода трассируемой программы из подпрограммы, а также индицированы состояние регистров МП и счетчика команд и флагов.

Процедуры трассировки — средства отладки прикладных программ, разрабатываемых с помощью ОСФФ, могут применяться в любых сочетаниях с вариантами команды G (см. команды подготовки и отладки прикладных программ). Однако при этом разработчику необходимо всегда помнить, что исполнение команды G с двумя или тремя адресными параметрами, а также директив трассировки (H и P) возможно только в случае, если отлаживаемая или трассируемая программа и подпрограмма, которыми она пользуется, находятся в области памяти, в которую МП может записывать информацию (в области памяти ОЗУ).

Команды формирования внешних файлов

В качестве носителя внешних файлов используется магнитная лента кассетного накопителя. Информация на ленту заносится непосредственно с экрана дисплея после вывода ее на экран директивой O.

OADDR1/ADDR2 — вывод данных на экран дисплея из области памяти, ограниченной адресами ADDR1 и ADDR2. Данные выводятся в 16-ричных числах в непрерывном формате без знаков сигнализации (AP-2 ВТА-2000—10). Первые два байта в выводимом массиве указывают начальный адрес области памяти, из которой данные выводятся. После вывода вся содержащаяся на экране информация может быть записана с помощью кассетного накопителя CM-5211 в одну страницу ленточного файла.

Телефон 23-63-75, Одесса

ЛИТЕРАТУРА

1. Дамке М. Операционные системы микроЭВМ.— М.: Финансы и статистика, 1985.
2. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоселектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.
3. Дирксен А. МикроЭВМ.— М.: Энергоиздат, 1982.
4. Гивоне Д., Россер Р. Микропроцессоры и микрокомпьютеры.— М.: Мир, 1983.

Статья поступила 17.02.88

ИЗМЕРИТЕЛЬНАЯ СИСТЕМА ДЛЯ АНАЛоговых и Аналого-цифровых интегральных микросхем

Разработка ПО контрольно-измерительных систем на языках высокого уровня не всегда приводит к оптимальному решению задач в прикладной области [1]. Для работы в реальном масштабе времени целесообразно использовать специальные языки программирования, позволяющие в диалоговом режиме подготавливать и отлаживать программы [1, 2].

Современная автоматизированная контрольно-измерительная система (АКИС) интегральных микросхем (ИМС) — сложный комплекс контрольно-измерительной аппаратуры и ПО, управляемый микроЭВМ.

С помощью МП К580ВМ80 создана компактная система с оптимальной архитектурой [3, 4], обслуживающая два рабочих места.

При разработке АКИС [5] использованы принципы модульного построения узлов и блоков и разделения АКИС на общую (базовую) и специализированную (модули и блоки для про-

верки конкретной ИМС) части. Базовый подход к проектированию обеспечил высокую эффективность разработки и эксплуатации АКИС и быструю адаптацию к условиям проверки новых типов ИМС, сохраняя максимальную «выживаемость» АКИС.

ПО программ реального масштаба времени [1, 2, 6], так же как и аппаратура АКИС, разделено на две части: общую (постоянную) — управляющую программу (УП), образующую ядро ПО, и специализированную (переменную) — данные (Д), описывающие процесс контроля конкретной ИМС.

УП размещена в РПЗУ АКИС и написана на ассемблере [7, 8]. При контроле ИМС блок обработки УП, состоящий из трех модулей (рис. 1), за один цикл выполняет следующие операции «типового» теста:

задает режимы работы ИМС и необходимую коммутацию цепей;

измеряет контролируемую величину или принимает код из порта чтения и

производит необходимые вычисления; оценивает допусковые значения измеренной величины.

Под «типовым» мы понимаем тест, на котором выполняются все три перечисленные операции.

УП при проверке одной микросхемы проходит столько рабочих циклов, сколько тестов содержат описания процесса контроля, и обеспечивает основные режимы работы АКИС: «ручной-автоматический», «измерение-контроль», «все параметры-до брака», «по параметрам-по тестам», «циклический-одиночный», «самоконтроль» и «регистрация», а также позволяет «заключивать» любой тест или параметр (рис. 1). Эти режимы хорошо зарекомендовали себя при отладке, метрологической аттестации, периодических проверках АКИС, разработке и исследовании ИМС. УП выдает на листинг дату, номер партии, число проверенных и процент выхода годных ИМС, результаты статобработки и гистограммы результатов измерений для конкретных условий производства ИМС [9].

Д описания процесса контроля, с которыми работает УП, подготавливаются пользователем в соответствующих форматах на «естественном» языке методом заполнения бланков.

При разработке общей части ПО АКИС целесообразно использовать языки высокого уровня, специальной части (получение машинных кодов) — метод заполнения бланков (рис. 2).

Система подготовки данных «Диалог 23», применяющая этот метод, обеспечивает удобный режим непосредственного общения разработчиков

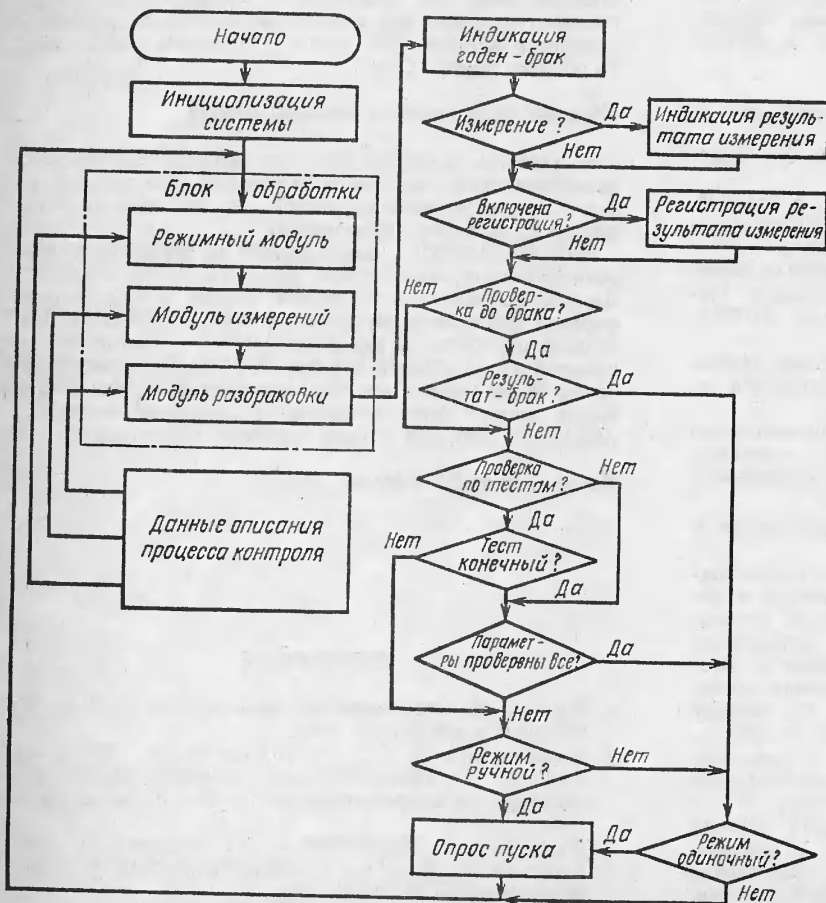


Рис. 1. Алгоритм обработки теста

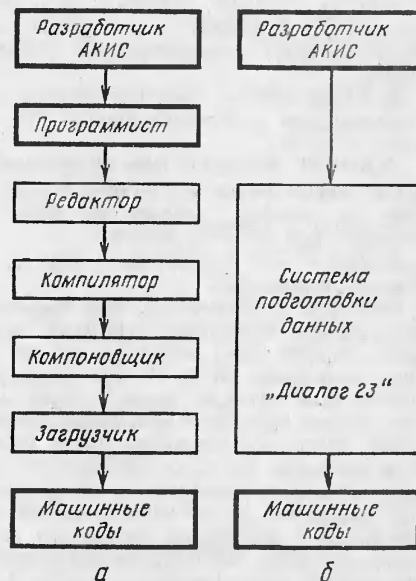


Рис. 2. Способы получения машинных кодов:

а — язык высокого уровня, б — метод заполнения бланков

АКИС с микроЭВМ в терминах русского языка [4]. ОС «Диалог 23» — расширение дисковой ОС — ОС1800 (аналог CP/M) [10, 11], хранится в виде файла на диске.

Система «Диалог 23» обладает всеми свойствами языков высокого уровня [1, 2] и автоматически переводит ответы пользователей в машинные коды, формат которых согласован с требованиями УП АКИС.

Выходные сообщения системы — бланки на естественном языке. Пользователь заполняет поля именами, числами или кодами, а система в режиме «подсказки» предлагает некоторые альтернативные варианты ответа. После окончания ввода одного элемента система автоматически перемещает курсор к началу следующего поля бланка. Все ответы пользователя переводятся системой в машинные коды и обрабатываются УП. Для разработки измерительной программы пользователю, как правило, достаточно иметь перечень применяемых блоков, их адреса и циклограмму описания процессора контроля ИМС [4].

В процессе отладки аппаратуры при необходимости изменения Д они вызываются с диска и редактируются.

Основные характеристики ПО АКИС:

Объем памяти, Кбайт	20
УП данных системы «Диалог 23»	8
Время подготовки данных, ч	1..4
Допустимое число:	
коммутационных точек	96
портов, доступных пользователю	160
контролируемых параметров	99
тестов в одном параметре	9
Время контроля одной ИМС, с	4..15
Тип используемой арифметики	С плавающей запятой

Общая часть ПО, разработанная в соответствии с базовым построением АКИС, остается постоянной и применяется во многих новых разработках

УДК 681.3

В. П. Семеренко, Я. А. Гуменюк

ПРОГРАММА РАСЧЕТА СИГНАТУР НА ПЕРСОНАЛЬНОЙ ЭВМ

При диагностировании микропроцессорных устройств широкое распространение получил метод сигнатурного анализа [1].

Экспериментальное определение тестируемости неисправностей цифровых устройств (ЦУ) путем их физического моделирования является очень трудоемким процессом, так как требует многократного снятия сигнатур в заданных точках схемы. Даже при однократном снятии сигнатур во всех точках схемы остается еще один важный недостаток — необходимость наличия заведомо исправного ЦУ. Поэтому актуальна разработка эффективных аналитических методов вычисления сигнатур [2]. Для расчета сигнатур часто используют аппарат циклических избыточных кодов. При этом необходимо выполнять сложную и громоздкую операцию деления двоичных последовательностей на характеристический полином сигнатурного регистра.

Вычисление сигнатуры может быть значительно упрощено, если в качестве теоретической основы сигнатурного анализа использовать математический аппарат линейной последовательностной машины (ЛПМ) [3]. В этом

контрольно-измерительной аппаратуры. Эта часть ПО обеспечивает общие функции системы: измерения, вычисления, компарирования, «зацикливания» тестов, вывода результатов обработки тестов на индикацию, печать и другие сервисные услуги. Как показал опыт разработки, Д, описывающие процесс контроля ИМС, занимают минимальный объем памяти при таком подходе построения ПО (предпочтительный тип памяти хранения Д — РПЗУ).

В систему подготовки данных «Диалог 23» входят модули упрощения ввода ошибочных или некорректных Д, поэтому процесс отладки измерительной программы исключается. Система позволяет разработчикам быстро и удобно создавать новые Д для контроля ИМС и редактировать их при отладке аппаратуры.

В настоящее время налажены мелкосерийное производство и промышленная эксплуатация АКИС. Затраты на разработку ПО по принципу деления на общую и специализированные части окупаются.

Недостаток рассматриваемого ПО — специфическая ориентация на базовую часть АКИС.

Телефон 442-95-94, Киев

ЛИТЕРАТУРА

1. Эйдукас Д. Ю., Орлов Б. В., Попель Л. М. и др. Измерение параметров цифровых интегральных микросхем / Под ред. Д. Ю. Эйдукаса, Б. В. Орлова. — М.: Радио и связь, 1982.
2. Денинг В., Эссиг Г., Маас С. Диалоговые системы «человек—ЭВМ». Адаптация к требованиям пользователя: Пер. с англ. — М.: Мир, 1984.
3. Григорьев В. Л. Программное обеспечение микропроцессорных систем. — М.: Энергоатомиздат, 1983.

4. Целуйко В. З., Климкович В. Г., Ратников В. Н., Пеньков И. В. Микропроцессорные системы управления серии «Гранит 02» // Микропроцессорные средства и системы. — 1987. — № 2. — С. 57—61.
5. Лившиц Д. Н., Котенко А. П., Захаренков С. Я., Белов В. И. Базовая автоматизированная система контроля для аналоговых интегральных микросхем, управляемая микроЭВМ // Механизация и автоматизация управления. — 1983. — № 3. — С. 43—45.
6. Мик Б., Хит П., Рашби Н. Практическое руководство по программированию: Пер. с англ. / Под ред. Б. Мика. — М.: Радио и связь, 1986.
7. Элфринг Г. Программирование на языке ассемблера для микроЭВМ: Пер. с англ. — М.: Радио и связь, 1987.
8. Попурый А. В. Система программирования ДИАЛОГ для микропроцессора КР5801К80 // Микропроцессорные средства и системы. — 1986. — № 2. — С. 28—32.
9. Лабунов В. А., Макушок Ю. Е., Пархутик В. П., Шершульский В. И. Диалоговая система сбора и статистического анализа данных для персональных компьютеров // Микропроцессорные средства и системы. — 1987. — № 2. — С. 32—33.
10. Гиглавый А. В., Кабанов Н. Д., Прохоров Н. Л., Шкамарда А. Н. МикроЭВМ СМ-1800: архитектура, программирование, применение. — М.: Финансы и статистика, 1984.
11. Уэйт М., Аргермейер Дж. Операционная система CP / М: Пер. с англ. — М.: Радио и связь, 1986.

Статья поступила 26.01.88.

случае n -разрядный l -входный сигнатурный регистр интерпретируется как ЛПМ, которая однозначно может быть описана с помощью матрицы A размерности $(n \times n)$ и матрицы B размерности $(n \times 1)$. Матрица A определяет структуру обратных связей сигнатурного регистра, а матрица B — структуру входных линий. Например, известный одноходовый 16-разрядный сигнатурный регистр с полиномом обратной связи $X^{16} + X^{12} + X^9 + X^7 + 1$ описывается следующими характеристическими матрицами A и B :

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}; B = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (1)$$

При воздействии на вход ЛПМ в момент времени t двоичного сигнала $u_i(t)$ ЛПМ перейдет из исходного состояния $S_{i-1}(t)$ в состояние $S_i(t)$, которое может быть определено по формуле

$$S_i(t) = A \cdot S_{i-1}(t) \oplus B \cdot u_i(t). \quad (2)$$

Формула (2) может быть также представлена в виде

$$S_i(t) = \begin{cases} A \cdot S_{i-1}(t) & , \text{ если } u_i(t) = 0, \\ A \cdot S_{i-1}(t) \oplus B, & \text{ если } u_i(t) = 1. \end{cases} \quad (3)$$

Очевидно, что сигнатура представляет собой код внутреннего состояния $S_i(t)$ ЛПМ, в которое ЛПМ перейдет после поступления последнего сигнала $u_i(t)$ двоичной последовательности с проверяемого вывода ЦУ.

Вычислить сигнатуры по формулам (2) или (3) достаточно просто, поскольку все матрицы в (2) и (3) являются булевыми и все операции сводятся к операциям сложения по модулю два. Например, если на вход сигнатурного регистра, который описывается матрицами A и B вида

$$A = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix}; \quad B = \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \end{vmatrix},$$

поступает последовательность сигналов $u_1(t)u_2(t)u_3(t) \times u_4(t) = 1001$, тогда сигнатура $S_4(t)$ будет получена на четвертом такте (исходное состояние $S_0(t)$ примем нулевым):

$$S_1(t) = A \cdot S_0(t) \oplus B = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix} \oplus \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 0 \end{vmatrix};$$

$$S_2(t) = A \cdot S_1(t) = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \end{vmatrix}; \quad S_3(t) = A \cdot S_2(t) = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \end{vmatrix};$$

$$S_4(t) = A \cdot S_3(t) \oplus B = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \end{vmatrix} \oplus \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 1 \\ 0 \\ 1 \\ 0 \end{vmatrix}.$$

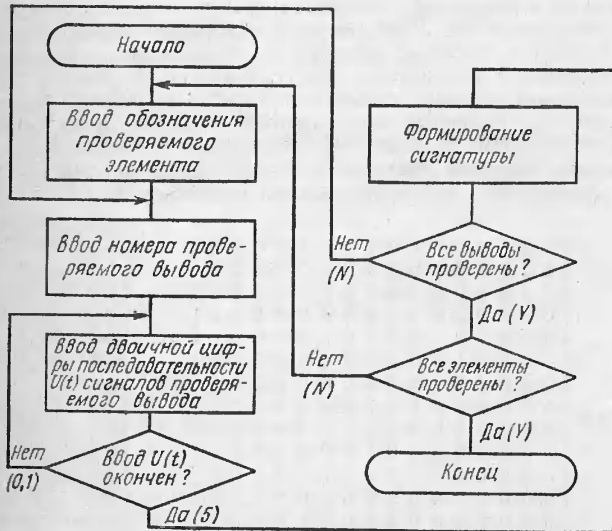
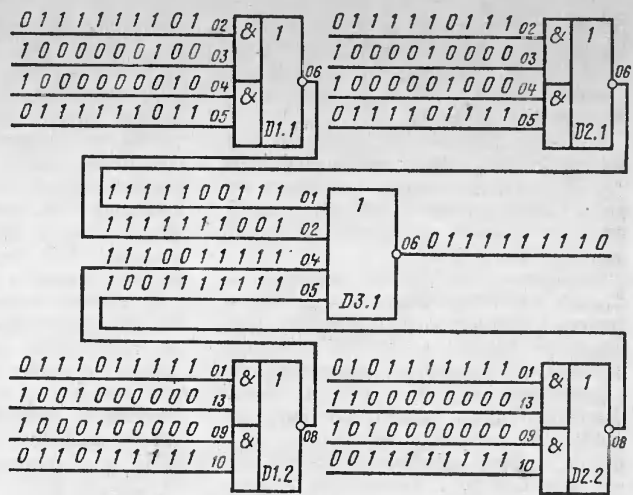


Рис. 1



D1, D2 - K155ЛР1; D3 - K155ЛА1

Рис. 2

На основе математического аппарата ЛПМ может быть составлена программа расчета сигнатур в автоматическом режиме в рамках САПР и в диалоговом режиме на персональной ЭВМ. На рис. 1 представлен алгоритм вычисления сигнатур в диалоговом режиме для цифровой схемы без обратных связей. Выполнение алгоритма состоит в последовательном вычислении сигнатур для всех выводов элементов схемы на основе предварительно определенных временных последовательностей сигналов в двоичной форме. На рис. 2 приведен пример схемы с требуемыми исход-

```

10 OPEN "LP:" FOR OUTPUT AS FILE #1
20 DIM A(16,16),B(16),S(16),U(25)
30 DIM T(16,16),U(16),P1(16),P2(16),C(16),P$(16)
75 N=16 \ L=1 \ D=1
80 DATA 0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,1
81 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
82 DATA 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
83 DATA 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
84 DATA 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
85 DATA 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
86 DATA 0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
87 DATA 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
88 DATA 0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
89 DATA 0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
90 DATA 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
91 DATA 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
92 DATA 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
93 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
94 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0
95 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
100 FOR I=1 TO N \ FOR J=1 TO N
110 READ A(I,J)
120 NEXT J \ NEXT I
130 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
140 FOR I=1 TO N \ READ B(I) \ NEXT I
160 DATA "0","1","2","3","4","5","6","7","8","9"
165 DATA "A","C","F","H","P","U"
167 FOR I=1 TO N \ READ P$(I) \ NEXT I
169 PRINT #1
170 PRINT "ВВЕДИТЕ ОБОЗНАЧЕНИЕ ПРОВЕРЯЕМОГО ЭЛЕМЕНТА"
175 INPUT H$
180 PRINT "СИГНАТУРЫ ДЛЯ ЭЛЕМЕНТА ",H$
185 PRINT #1," СИГНАТУРЫ ДЛЯ ЭЛЕМЕНТА ",H$
190 FOR I=1 TO N \ S(I)=0 \ NEXT I
214 PRINT H$
220 PRINT "ВВЕДИТЕ ДВУЗНАЧНЫЙ НОМЕР ПРОВЕРЯЕМОГО ВЫВОДА"
225 INPUT N1$
227 E=0
230 PRINT "ВВЕДИТЕ ЧИСЛОВУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ"
235 PRINT "В КОНЦЕ ЧИСЛОВЫХ ДАННЫХ НАВЕРНИТЕ ЦИФРУ 5"
237 T=1
240 INPUT U(1)
247 IF U(1)=0 THEN 254
248 IF U(1)=1 THEN 254
250 IF U(1)=5 THEN 260
251 PRINT "НЕПРАВИЛЬНАЯ ЦИФРА" \ GO TO 240
254 I=I+1 \ E=E+1 \ GO TO 240
260 PRINT "ВВОД ОКОНЧЕН"
310 FOR K=1 TO E
320 M1=N
330 FOR I=1 TO N \ FOR Z=1 TO N
340 T(I,Z)=A(I,Z)

```

Рис. 3 (Начало)

```

350 NEXT Z \ NEXT I
360 PRINT
370 FOR I=1 TO M1 \ W(I)=S(I) \ NEXT I
400 GOSUB 1000
410 FOR I=1 TO N \ P1(I)=C(I) \ NEXT I
440 M1=L
450 FOR I=1 TO N \ T(I,1)=B(I) \ NEXT I
480 FOR Z=1 TO M1 \ W(Z)=U(K) \ NEXT Z
510 GOSUB 1000
520 FOR I=1 TO N \ P2(I)=C(I) \ NEXT I
550 FOR I=1 TO N
560 IF P1(I)=P2(I) THEN 580
570 S(I)=1 \ GO TO 590
580 S(I)=0
590 NEXT I
592 PRINT "K=";K
595 NEXT K
597 PRINT "ВЫВОД ";N1$;
598 PRINT #1, "      ВЫВОД ";N1$;
600 FOR Z=1 TO N-3 STEP 4
610 W=S(Z)*#B+S(Z+1)*#4+S(Z+2)*#2+S(Z+3)*#1+1
625 PRINT P$(W);
626 PRINT #1,P$(W);
630 NEXT Z
645 PRINT
650 FOR I=1 TO N \ PRINT B(I); \ NEXT I
680 PRINT #1
700 PRINT "РАСЧЕТ СИГНАТУР ДЛЯ ДАННОГО ЭЛЕМЕНТА ОКОНЧЕН?Y/N"
710 INPUT F1$
720 IF F1$="N" THEN 190
730 PRINT "ПРОДОЛЖИТЬ РАСЧЕТЫ ДЛЯ СЛЕДУЮЩЕГО ЭЛЕМЕНТА?Y/N"
740 INPUT F2$
750 IF F2$="N" THEN 1210
760 D=D+1 \ GO TO 169
770 GO TO 169
1000 FOR I=1 TO N
1010 R1=0
1020 FOR H=1 TO M1
1030 R2=T(I,H)*#W(H)
1040 IF R1=R2 THEN 1070
1050 R1=1 \ GO TO 1100
1070 R1=0
1100 NEXT H
1110 C(I)=R1
1120 NEXT I
1130 RETURN
1210 CLOSE #1
1220 END

```

Рис. 3 (Скоп.)

ными данными: тестовыми наборами на входных полюсах схемы и соответствующими значениями сигналов во внутренних и выходных полюсах схемы.

Указанный алгоритм вычисления сигнатур реализован в виде программы на языке БЕЙСИК для персональной ЭВМ ДВКЗ (рис. 3). В начальной части программы задаются матрицы А и В вида (1) и начальное нулевое состояние $S_0(t)$. Затем согласно алгоритму вводятся обозначение проверяемого элемента, номер очередного проверяемого вывода элемента и двоичная последовательность сигналов $u(t)$. Приведенные в скобках обозначения возле условных операторов алгоритма (см. рис. 1) соответствуют возможным вариантам ответа пользователя на предусмотренные в программе вопросы.

Основная часть программы является реализацией формулы (3); в ней имеются обращения к подпрограмме

СИГНАТУРЫ ДЛЯ ЭЛЕМЕНТА		K155LP1
ВЫВОД 02		D1.1
ВЫВОД 03		3U40
ВЫВОД 04		0100
ВЫВОД 05		F080
		9PF0
ВЫВОД 09		D1.2
ВЫВОД 10		17F0
ВЫВОД 01		9000
ВЫВОД 13		8800
		0UF0
ВЫВОД 02		D2.1
ВЫВОД 03		1HF0
ВЫВОД 04		8400
ВЫВОД 05		8200
		1CF0
ВЫВОД 09		D2.2
ВЫВОД 10		3UF0
ВЫВОД 01		F000
ВЫВОД 13		A000
		SUF0
ВЫВОД 01		D3.1
ВЫВОД 02		99F0
ВЫВОД 04		5P40
ВЫВОД 05		87F0
ВЫВОД 06		UUF0
		CU00

Рис. 4

булевого перемножения двумерной матрицы на вектор (рис. 4).

Телефон 4-63-66, Винница

ЛИТЕРАТУРА

1. Гордон Д., Надиг Х. Локализация неисправностей в микропроцессорных схемах с помощью 16-ричных ключевых слов // Электроника.— 1977.— № 5.— С. 23—33.
2. Ярмолик В. Н. Аналитический метод вычисления сигнатур для сетевых дискретных структур // Автоматика и вычислительная техника.— 1987.— № 5.— С. 77—81.
3. Гилл А. Линейные последовательностные машины.— М.: Наука, 1974.— 288 с.

Статья поступила 3.05.88

УДК 681.3:657.47

Е. Ю. Каштанов, А. Е. Корчак

ПОДСИСТЕМА ПЕЧАТИ ДОКУМЕНТАЦИИ «ПРИН» ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

Одним из важнейших применений персональных компьютеров (ПК) в народном хозяйстве является совершенствование информационной деятельности и делопроизводства. Для подготовки и тиражирования управленческих документов (письма, справки, отчеты и др.) и научно-технической документации (обзоры, статьи, монографии) с помощью ПК используют системы подготовки документации (СПД), которые значительно облегчают и ускоряют процесс набора, редактирования и копирования документов по сравнению с безмашинной технологией.

Для 8-разрядных ПК типа Роботрон-1715, Корвет, СМ1800 и других наиболее популярны СПД, аналогичные текстовому процессору WordStar [1], такие как РЕФОР [2], ТЕКСТ [3], «Документ-2» [4] и др. Эти СПД предоставляют развитые средства по набору и редактированию текстов документов в экранном режиме, печати документов на бумаге с выделением участков текста другим шрифтом или цветом, управлению вертикальным и горизонтальным размещением текста на странице. Имеется средство (MAILMERGE) для печати однотипных писем, справок со стандартным текстом, дополняемым переменной информацией из некоторого файла.

Стандартные средства СПД не позволяют печатать документы сложной структуры (книги, программная и другая техническая документация), имеющие рисунки и таблицы, требующие генерации содержания и предметного указателя. Для их подготовки были разработаны системы WordIndex II фирмы Borland [5] и StarIndex фирмы MicroPro [6]. Опыт эксплуатации этих систем выявил следующее:

WordIndex II содержит ошибки, в частности неверно нумерует страницы в содержании и указателе;

обе системы являются предпроцессорами для СПД WordStar и из исходного текста документа формируют файл, дополненный заголовками рубрик, содержанием, указателем. Полученный файл выводится на печать стандартными средствами СПД WordStar. Для ПК с гибкими дисками использование этих систем затруднено при работе с документами большого объема;

не допускаются многострочные верхние и нижние заголовки (требование ЕСПД) и названия рубрик, рисунков, таблиц;

нельзя задать сортировку предметного указателя по русскому алфавиту.

Разработана подсистема печати документации, предоставляющая все возможности по управлению печатью, имеющиеся в СПД WordStar, а также: вывод документа на печать или в дисковый файл с автоматической

нумерацией рубрик, рисунков, таблиц, созданием содержания, списка рисунков и таблиц; автоматическое составление предметного указателя с заголовками и подзаголовками, отсортированными по русскому или латинскому алфавиту; обработку текстов, заданных разными способами кодирования символов; включение в текст графических изображений (рисунки, графики). Поддерживается шесть видов кодирования; пять 7-битных (КОИ-7/0 (латинский) КОИ-7/1 (русский), КОИ-7/2 (прописные латинские и русские буквы), два набора, КОИ-7/0 и КОИ-7/1 с управляющими кодами переключения наборов (^E, ^P или ^N, ^O)) и один 8-битный (КОИ-8).

После запуска подсистемы запрашивается имя входного файла, направление вывода (на печать или в файл), номера страниц, подлежащих печати (отдельные номера или интервалы), вывод всех страниц или только с четными или нечетными номерами (на двух сторонах непрерывной ленты бумаги), печать по листам или непрерывная, с форматизацией или без нее, количество копий. Команды СПД WordStar расширены возможностью задания разных параметров для страниц с четными и нечетными номерами (верхний и нижний заголовок, смещение относительно левого края листа).

Документ может быть разделен на рубрики пяти ступеней. Для рубрики задается система нумерации (порядочная или индексационная, содержащая номера текущих рубрик старших ступеней, например 2.1.2), вид нумерации (арабскими или римскими цифрами, латинскими или русскими буквами, либо без нумерации), обозначение рубрики (глава, раздел), шрифт. Допускается нумерация абзацев, не имеющих заголовков. Из названий рубрик формируется содержание, включающее ссылки (номера страниц) на рубрики. Для рисунков и таблиц генерируются заголовки, содержащие порядковые номера, их название включается в список рисунков или таблиц. Допускается размещение таблицы или рисунка на нескольких страницах с обозначением их отдельных частей.

Для создания предметного указателя в тексте выделяют слова или словосочетания с помощью управляющих символов или задают их в командах. Может быть выделена и ссылка на страницу с наиболее важной информацией о предмете (определение, выводы). Для каждого заголовка указателя формируется перечень страниц (адресные ссылки), на которых он встречается. Вместо адресной ссылки (номер страницы) может использоваться отсылка, т. е. переадресовка к заголовку с адресной ссылкой, например компьютер, см. ЭВМ.

Заголовок указателя может иметь подзаголовки:

ЭВМ, 5, 24, 28 — заголовок,
мини, 8, 12 — подзаголовок,
персональная, 5, 14 — подзаголовок.

Подсистема печати работает под управлением операционных систем CP/M и МикроДОС с ОЗУ не менее 40 Кбайт. Поставляется версия, настроенная на «Роботрон-1715» с принтером K6312 (СМ6329.01М). Обеспечивается настройка на другие типы принтеров. Подсистема печати разработана на языке программирования BDS C [7].

Подсистема печати входит в состав системы редактирования и подготовки документации «Документ-2», которую можно приобрести для ПЭВМ «Роботрон-1715», направив гарантийное письмо по адресу: 226063, Рига, ул. Кенгарае, 10, ЛАТВИИ; тел 36-36-44. Рига

ЛИТЕРАТУРА

1. WordStar, Reference Manual. WS-3550-1 (3.0).— MicroPro International Corp., 1981.—151 p.
2. Система подготовки текстов «Рефор».— Таллинн: НУЦ, 1987.
3. Система подготовки документации ТЕКСТ // Методические материалы и документация по пакетам

прикладных программ.— М.: МЦНТИ, 1987.— Вып. 42.— 70 с.

- 4: Документ-2 // Каталог диалоговых систем: Материалы по математическому обеспечению.— Киев: Ин-т кибернетики им. В. М. Глушкова АН УССР, 1986.— С. 16.
5. WordIndex-11. User's Manual.— Borland Ltd, 1981.— 70 p.
6. StarIndex. Reference Manual. Release 1.0.— MicroPro International Corp., 1983.— 286 p.
7. C Compiler v. 1.5. User's Guide.— BD, Software, 1982.— 190 p.

Статья поступила 21.01.88

УДК 681.3:621.3

В. В. Новиков, А. В. Орехов

ИНТЕРАКТИВНАЯ ТРАССИРОВКА ПЕЧАТНЫХ ПЛАТ НА ДВКЗМ2

Для микроЭВМ ДВКЗМ2 разработан комплекс программ, обеспечивающих интерактивную трассировку двухслойных печатных плат. Пользователь разводит плату вручную, используя графический редактор. Затем вспомогательные программы готовят управляющие перфокарты для технологического оборудования. Программное обеспечение (ПО) состоит из следующих компонентов: редактора топологии печатных плат; программы для подготовки перфокарты во входном формате фотоплоттера (например, типа ЕММА2 фирмы «Квест»); программы для подготовки перфокарты во входном формате сверильного станка с ЧПУ (например, марки ВР910РЭ); программы для прорисовки изображения топологии на матричном АЦПУ (например, типа СМ6325).

Редактор топологии — это графический редактор, работающий с графическими примитивами: контактной площадкой, отрезком проводящей дорожки, алфавитно-цифровым (АЦ) символом (КОИ-7, набором Н2). Контактные площадки и проводники рисуются в соответствии с размерами, выбираемыми пользователем из множества разрешенных. Размеры определяются тем, для какого слайда (носителя диафрагм на фотоплоттере ЕММА) создается изображение. Номер слайда пользователь указывает в начальном диалоге. Описание слайда загружается в редактор из специального файла. При таком подходе к прорисовке пользователь может визуальную контролировать правильность создаваемой топологии.

Редактирование изображения идет на активном слое. Другой слой пассивный, но может быть виден (по желанию пользователя). Команды редактора осуществляются нажатием на функциональные клавиши (при разработке претерпевали принцип «Человек в мире объектов»)*. Общее число команд больше 40. Раскладка команд по клавишам приведена ниже. В редакторе топологии реализованы два режима: РЕДАКТОР и ПИШУЩАЯ МАШИНКА. Режим ПИШУЩАЯ МАШИНКА предназначен для создания надписей на плате. В этом режиме можно воздействовать только на АЦ символы, при этом возможны дополнительные команды типа «Сдвинуть на знакоместо», «Сдвинуть на строку». АЦ символы можно рисовать в четырех ориентациях (параллельно каждой стороне платы).

РАСКЛАДКА КОМАНД ПО КЛАВИАТУРЕ ДИСПЛЕЯ 15IE-00—013

<^N>	— Дать список команд
<^E>	— Закончить работу
<СТРЕЛКИ>	— Управление курсором
<^>= <СТР>	— Установить шаг курсора (с дискретом 0,01 мм)

* Лебедев Г. В. Разработка интерактивных программ на основе принципа непосредственного редактирования информации // Микропроцессорные средства и системы.— 1986.— № 1.— С. 44—46, 58.

< ^K > = < CTC >	— Сменить тип курсора
< ПРОБЕЛ >	— Удалить символ
< HOME >, < ПРОБЕЛ >	— Удалить символы, принадлежащие фрагменту
< ^Y >	— Изменить активный слой
< ^W >	— Изменить видимость слоев
< SEND >	— Совместить центр окна с курсором
< ^I > = < GT >	— Увеличить масштаб
< ^X > = < TAB >	— Уменьшить масштаб
< HOME >, < СТРЕЛКА >	— Курсор к границе окна
< ^R >	— Прочитать из файла
< ^T >	— Записать в файл
< HOME >, < R >	— Прочитать фрагмент из файла
< HOME >, < T >	— Записать фрагмент в файл
< ^Z >	— Установить угол поворота фрагмента
< ^F > = < PRС >	— Скопировать фрагмент
< ^_ > = < ПРД >	— Передвинуть фрагмент
< ^L > = < СБП >	— Перерисовать экран
< ^U >	— Удалить все символы
< ^N > = < РУС >	— Разрешить/запретить рисование шкалы
< ^P > = < АП >	— Включить/отключить зеркальное преобразование
< A > = < C1 >	— Режим РЕДАКТОР / Режим ПИШУЩАЯ МАШИНА

В режиме РЕДАКТОР

< S >	— Задать размеры платы
< P >	— Поставить площадку
< A >	— Сменить размеры площадки
< W >	— Запретить/разрешить вывод меню площадок
< M >	— Начинать линию от курсора
< L >	— Провести линию до курсора
< D >	— Сменить толщину линии
< U >	— Запретить/разрешить вывод меню линий
< T >	— Поставить крепежное отверстие
< K >	— Сообщить значения параметров редактора

В режиме ПИШУЩАЯ МАШИНА

< ^M > = < BK >	— Курсор в начало следующей строки
< ^J > = < PC >	— Курсор на следующую строку
< IC >	— Строку справа от курсора — вправо на знакоместо
< DC >	— Строку справа от курсора — влево на знакоместо
< DL >	— Текст ниже курсора — вниз на строку
< IL >	— Текст ниже курсора — вверх на строку
< ^G >	— Установить ориентацию АЦ символов

Обозначения

< ^X >	— Одновременное нажатие клавиш < CY > и < X >
< HOME >	— Стрелка влево вверх
< SEND >	— Стрелка вниз с точкой
< IC >	— Стрелка вправо с чертой (вставка символа)
< DC >	— Стрелка влево с чертой (удаление символа)
< IL >	— Стрелка вверх с чертой (вставка строки)
< DL >	— Стрелка вниз с чертой (удаление строки)

Так как в ДВКЗМ2 используется монохромная графика, то первый слой рисуется непрерывной линией, а второй слой — штриховой. Для удобства работы на втором слое можно сделать зеркальное преобразование изображения относительно вертикальной оси. В редакторе есть механизм работы с фрагментом изображения. Под фрагментом редактор понимает часть изображения, ограниченную произвольным прямоугольником. Границы фрагмента пользователь указывает с помощью курсора. Если перед операцией по копированию или перемещению фрагмента был задан отличный от нуля угол поворота, то фрагмент еще и поворачивается.

Из штатного оборудования в ДВКЗМ2 установлена плата контроллера перфоратора (плата В2 1). Кроме того, штатные монитор Электроника МС6105.1, клавиатура и плата КСД заменены на дисплей 15ИЭ-00—013. Эта замена освободила место в блоке сопряжения ДВК. Первоначально в качестве внешней памяти редактор использовал НГМД. Для увеличения быстродействия редактора применили внешнюю электронную память, соединив ДВК с микроЭВМ «Электроника ДЗ 28» (ОЗУ 128 Кбайт) с помощью модифицированной платы И2. Разрабатывается плата ОЗУ (128 Кбайт) размером в полплаты формата ДВК. Редактор настраивается на любую электронную память, доступную через двунаправленный порт ввода-вывода. Дополнительной памяти 128 Кбайт редактору достаточно для разводки полной платы формата ДВК со 100 корпусами по 16 выводов. Максимальные размеры платы, которую можно развести в редакторе, составляют 640×640 мм.

Описанное ПО кроме редактора топологии используется как эмулятор сколки (диджитайзера) в системе производства печатных плат, поставляемой фирмой «Квест». В настоящее время разрабатывается программа для конвертирования перфолент, выпускаемых системами автоматической трассировки типа ПРАМ и РАПИРА на ЕС ЭВМ, в графический файл редактора топологии. Применение ПО возможно и вместо кодировщиков в системе МОТИФ*. Все ПО написано на Фортране, поэтому возможно быстрое перенесение его на другие микроЭВМ или настройка на другое технологическое оборудование.

Телефон 535-08-72, Москва

Статья поступила 22.01.88

* Алексеева Р. П., Жильцов А. В., Коробова Л. Е., Михайлова О. А., Песков М. И., Суляцкий Ю. Н. ДВК-2М в производстве фотоматричных печатных плат // Микропроцессорные средства и системы. — 1987. — № 2. — С 50—52.

РЕКЛАМА

Смоленское ПО «Искра» предлагает современные печатающие устройства (принтеры) моделей МТ-290 и МТ-81 совместного производства ПО «Искра» и фирмы «Маннесманн Талли», ФРГ. Принтеры могут применяться при работе с персональными ЭВМ и других работах, связанных с выводом информации.

Изделия представляют собой точечно-матричные устройства последовательного вывода.

Основные технические данные:

Средняя наработка на отказ: МТ-290 — 200 млн. зн.; МТ-81 — 30 млн. зн.

Интерфейсы: типа «Центроникс»;

Тип печати: обычная и улучшенного качества;

Темп печати: МТ-290 — 200 зн/с, МТ-81 — 130 зн/с;

Число копий: МТ-290 — оригинал+4 копии,

МТ-81 — оригинал+2 копии

Потребляемая мощность: МТ-290 — 100 ВА, МТ-81 — 40 ВА;

Масса: МТ-290 — 11 кг, МТ-81 — 4 кг.

Оплата стоимости производится частично в валюте первой категории.

За справками обращаться по адресу: 214020, Смоленск, ул. Бабушкина, 1, ПО «Искра». Тел. 1-33-75 и 1-33-21.

УДК 681.3.06

Е. А. Жарков, В. И. Ковальцов, Г. М. Максимов

КРОСС-СИСТЕМА ПРОГРАММИРОВАНИЯ 8- И 16-РАЗРЯДНЫХ МИКРОПРОЦЕССО- РОВ С УНИФИЦИРОВАННЫМ ФОРМАТОМ ОБЪЕКТНЫХ МОДУЛЕЙ

Быстрое развитие микропроцессорной техники накладывает отпечаток на развитие программного обеспечения (ПО) для микропроцессоров (МП), определяя преемственность разработок не только за счет аппаратных решений (совместимости снизу вверх, как, например, для 8- и 16-разрядных микропроцессоров фирмы Intel), но, главным образом, за счет унификации и стандартизации средств программирования. Все большее распространение получают языки программирования микроЭВМ высокого и среднего уровней, реализуемых зачастую на универсальных ЭВМ в виде кросскомпиляторов (PLM, Паскаль, Си и др.) [1, 2]. Таким образом, достигается определенная независимость прикладного (и даже системного) ПО от типа микроЭВМ. Кросскомпиляторы для этих языков, в свою очередь, разрабатываются на некотором широко распространенном языке (например, FORTRAN, BCPL, Си), что обеспечивает переносимость кросскомпилятора на многие существующие универсальные ЭВМ [3, 4]. Следовательно, можно добиться независимости от типа базовой ЭВМ, на которой создается кроссовое ПО. Но это лишь одна сторона проблемы. Не менее важным является вопрос о степени зависимости кросс-системы от целевой микроЭВМ, т. е. ее архитектуры, разрядности, организации интерфейса с внешними устройствами и пр. Очевидно, компиляторы должны в полной степени учитывать эти особенности исполнительской системы. Однако введением промежуточной машинно-независимой формы представления исходной программы (на языке абстрактной машины) удастся выделить в компиляторе процедуры лексического, синтаксического и семантического анализа, независимые от деталей исполняемого кода. При этом вся зависимость от конкретного МП переносится на этап трансляции языка абстрактной машины в коды МП. Значит, при переходе на другую микроЭВМ достаточно переписать лишь генератор кодов, что в несколько раз упрощает реализацию кросскомпилятора [5].

Подобный подход можно применить и на нижнем уровне иерархии кросс-системы программирования, а именно в структуре объектных модулей (объектный модуль — представляемые коды и данные и дополнительная информация о внешних связях, которые относительно просто преобразуются в исполняемые коды, эквивалентные исходной программе). В общем случае формат объектного модуля уникален для каждой ЭВМ, да и в различных системах ПО для одного и того же типа ЭВМ. Для микроЭВМ и кроссового ПО такое решение будет неудовлетворительным из-за особенностей разработки ПО для МП. Каждая пара — базовая ЭВМ и микроЭВМ — в таком случае определяли бы свое уникальное ПО, что весьма нерационально.

Структура и назначение CUFOM-системы

Подобная проблема исследовалась в ЦЕРН (CERN), в результате чего был выработан единый формат объектных модулей CUFOM (the Universal Format for Object Modules) [6]. В CERN разработано кроссовое ПО [7], функционирующее на трех базовых ЭВМ (NORD-10, IBM, CDC) и обслуживающее несколько типов МП: Intel

8080, Motorola 6800, TMS 9900, FI00-L. Мини-компьютеры типа PDP-11/45, NP-2100 и MODCOMP-11 связываются через CERNET для разработки ПО на базовых ЭВМ и собственных средств программирования МП не имеют. Кросс-система включает в себя трансляторы для соответствующих микроЭВМ, производящие объектные модули в CUFOM-формате, универсальные редактор связей и библиотечарь объектных модулей, а также средство загрузки готовых к исполнению программ — настраивающий предзагрузчик. Редактор связей, библиотечарь, транслятор и предзагрузчик называют CUFOM-процессором для некоторой микроЭВМ [8].

В Институте физики высоких энергий (ИФВЭ) на текущий момент доступны два CUFOM-процессора: для 8-разрядной микроЭВМ, построенной на базе МП серии K580, и для 16-разрядной микроЭВМ — на базе МПК БИС K1810. CUFOM-процессоры реализованы для мини-ЭВМ CM4 и ЭВМ DEC-10 и полностью совместимы для этих машин [3, 4]. Кроме того, имеется транслятор для МП K580, генерирующий сразу абсолютный загружаемый модуль в формате предзагрузчика для этого МП, что позволяет использовать их в единой кросс-системе ПО МП. Что касается языков высокого уровня, то для 8-разрядного МП разработан кросскомпилятор с языка PL/H, а для 16-разрядного МП — с языка PL/M-86. Оба компилятора на выходе дают программу на ассемблере соответствующей микроЭВМ и, таким образом, не привязаны к какому-либо формату собственно объектных модулей в кросс-системах ПО.

На завершающей фазе работы CUFOM-процессора (при подготовке объектного файла к виду, готовому к непосредственной загрузке в память микроЭВМ и исполнению) модуль уже не содержит внешних ссылок и по указанию базового адреса преобразуется в абсолютную программу, которая загружается с помощью резидентного загрузчика по линии связи микроЭВМ с базовой машиной.

Редактор связей и библиотечарь объектных модулей в CUFOM-системе не зависят от типов базовой ЭВМ (CM4, DEC-10) и микроЭВМ вследствие принятого подхода к разработке CUFOM-системы [6] и выбора переносимого языка программирования BCPL [14], обеспечивающего символьное представление CUFOM-модулей (отсюда и возможность транспортировки по линиям связи, как, например, в сети INDEX). Независимость от типа микроЭВМ определяется структурой CUFOM-модуля и неразрывно связана с ней. Редактор связей позволяет разработать прикладное ПО на различных языках и состыковать объектные модули на этапе разрешения внешних связей. Совокупность модулей может быть организована в библиотеку с помощью программы-библиотечаря объектных модулей.

Универсальность этих двух элементов кросс-системы программирования МП рассматривалась разработчиками CUFOM-системы CERN как наиболее важное проектируемое свойство. Набор базовых директив, используемых в CUFOM-модуле, невелик, хотя и описывает полностью структуру модуля. Возможности CUFOM-процессоров для конкретной микроЭВМ (если это необходимо) расширяются вызовами машинно-зависимых функций, в то время как сам способ обращения к таким функциям (а лишь это важно для редактора связей или библиотечаря) стандартен для всех CUFOM-процессоров. Структуры же объектных модулей для различных CUFOM-процессоров схожи, что предопределяет преемственность даже их машинно-зависимых элементов. Тем самым обеспечивается переносимость, например предзагрузчика, в той его части, которая анализирует структуру модуля, и переработки требует лишь генератор двоичных кодов, очевидно, ориентированный на длину машинного слова, организацию памяти и т. п.

В первую очередь, конечно, преемственность ПО CUFOM-системы минимизирует усилия и затраты, связанные с разработкой и реализацией системы программирования МП на другой базовой ЭВМ или при выборе другого МП. С точки зрения пользователя, существует стандартный набор сервиса для любой комбинации базовых и микро-ЭВМ: документация, интерфейс, процедуры и алгоритмы. Изучение нового CUFOM-процессора для пользователя сводится лишь к уточнению машинно-зависимых свойств CUFOM-процессора. Важным фактором является и быстродействие CUFOM-системы: модули имеют простую структуру, детали машинного представления программ вводятся функциями со стандартным интерфейсом и откладываются до фазы предзагрузки. Кроме того, этап редактирования связей может быть опущен (для простых модулей, не имеющих внешних связей), так как отредактированный модуль представляется в том же CUFOM-формате; возможно также частичное редактирование связей, а его результат наравне с другими объектными модулями может быть подвергнут дальнейшим операциям редактирования.

ПО для ОЭВМ серии КР1816

Программирование ведется на языке ассемблера [1], транслятор с которого (I48) разработан на базовых ЭВМ СМ4 и DEC-10. Обе версии языка полностью совместимы, языком реализации служит BCPL. Объектный модуль, получаемый транслятором I48, имеет формат, принятый в CUFOM-системе для загрузочных модулей. Таким образом, отсутствуют фазы редактирования связей и предзагрузки, а модуль представляется в виде абсолютной программы, готовой к исполнению.

О реализации кросстраницаторов для МП серии К580

Язык ассемблера ASM80 полностью соответствует описанному в работе [9]. Однако были сделаны некоторые дополнения [3], в частности возможность выделения комментариев символом «;» в начале строки или в поле комментария (в отличие от пробела). Интерфейс пользователя с программой (т. е. способ задания входных и выходных файлов и ключей) определяется той операционной средой, в которой она работает, и должен максимально соответствовать принятым в этой среде стандартам.

В кросстраницаторе реализованы все возможности классических макроассемблеров: условная трансляция, макросредства, аппарат внешних имен, перемещаемость производимой объектной программы (если не задано противоположное), включение других файлов в момент трансляции посредством директивы INC (INCLUDE). Поэтому при переносе программы на 16-разрядную мини-ЭВМ СМ4 с ограниченной памятью структура программы реорганизована в оверлейную (т. е. с динамически загружаемыми сегментами; постоянно в оперативной памяти ЭВМ находится лишь корневого сегмент). Это обеспечило значительное адресное пространство для внутренних таблиц транслятора. Динамически загружаются процедура анализа командной строки, оба прохода трансляции и макробработка.

В некоторых приложениях для разработки небольших программ или при недостаточной мощности базовой ЭВМ (например, микроЭВМ «Электроника 60») требуется простое и компактное кроссовое ПО. Вместо принятого в CUFOM-системе процесса «трансляция в объектную программу — редактирование связей — предзагрузка» здесь необходимо осуществить только первый этап — трансляцию. Для этих целей разработан язык ассемблера I80 [3], реализующий минимум возможностей управления трансляцией (директивы SET, EQU, ORG, DB, DW, RES) со скромными потребностями в оперативной памяти (около 18 Кслов). Производимый им объектный файл имеет формат загрузочного модуля в CUFOM-системе.

Загружаемая в ОЗУ МП программа может быть предварительно отлажена с помощью диалогового интерпретатора команд МП-программы IMI [10].

Входной формат имитатора согласован с выходом предзагрузчика PLD80 и транслятора I80.

Для программирования МП К580 применяется язык PL/H [3]. Компилятор PL/H строит объектный модуль в виде программы на ассемблере, которая в дальнейшем обрабатывается транслятором ASM80 или I80. Модуль может содержать внешние ссылки, но не является перемещаемым и размещается с указанного в команде ORIGIN абсолютного адреса (вставкой директивы REL модуль становится перемещаемым, но область данных остается абсолютно размещаемой).

Разработка CUFOM-процессора для МП серии К1810

Для программирования 16-разрядного процессора в ИФВЭ применяются кросскомпиляторы с алгоритмических языков PL/M-86 и ASM86 [4]. Возможность включения 16-разрядного МП в единую систему ПО для МП представлялась авторам весьма привлекательной, и такая реализация на практике еще раз подтвердила правильность подхода, принятого в CUFOM-системах.

Доступность разнообразных универсальных программных средств (редактор связей, библиотечка объектных модулей) предопределила работу авторов по расширению имеющейся реализации [4] и доведения ее до полного соответствия со стандартом фирмы Intel на этот язык [11]. Объектный файл, генерируемый программой ASM86, представляется в CUFOM-формате. Используются лишь стандартные возможности этого формата, а машинно-зависимые факторы определяются введением нескольких вызовов функций (US) для задания характеристик сегментов: способа выравнивания границ сегмента, базового адреса для абсолютного сегмента, спецификаций сегментов по типу памяти, описания сегментов, объединенных директивой ассемблера GROUP, и др.

В модифицированной версии кросстраницатора было решено отказаться от библиотеки ввода-вывода FORTRAN-компилятора; написан модуль, реализующий необходимые операции ввода-вывода. Размер задачи около 28 Кслов. Объектный модуль для загрузки его в исполнительную систему (МП К1810 и монитор SDK-86 [12]) обрабатывается предзагрузчиком PLD86. Распределение перемещаемых сегментов в памяти полностью управляется предзагрузчиком; можно выбрать любой порядок следования сегментов. Программа PLD86 написана на языке Си для ЭВМ СМ4.

Язык программирования PL/M-86: реализация и ограничения

Одним из первых языков программирования высокого уровня для МП Intel 8086 стал PL/M-86 [13]. Язык разрабатывался исключительно для данного процессора, хотя и сохранил преемственность к своему предшественнику PL/M-80 [2]. Последний возник вместе с МП Intel 8080 как подмножество языка PL/1, но с рядом ориентированных на структуру МП особенностей: тип данных (BYTE, WORD — целое без знака, 8 и 16 бит соответственно; ADDRESS — 16-разрядное целое, представляющее физический адрес микроЭВМ); управление памятью (статическое и динамическое распределение памяти, возможно обращаться к данным по физическому адресу через указатели); обработка исключительных ситуаций (определение процедур обработки прерываний по любому из восьми уровней, запрещение и разрешение прерываний из программы пользователя); оператор останова процессора HALT; управление вводом-выводом (операторы INPUT, OUTPUT с 8-разрядными портами внешних устройств); доступ к регистрам-признакам (функции CARRY, SIGN, ZERO, PARITY) и указателю стека (STACKPTR).

С. В. Пашенков, М. Ф. Стерин

РАЗРАБОТКА ПРОГРАММ ДЛЯ МИКРОПРОЦЕССОРА K580IK80 С ИСПОЛЬЗОВАНИЕМ КРОСС-СИСТЕМЫ i85

В определении языка PL/M-86 были сохранены все перечисленные выше особенности, так как они в равной степени переносились от 8-разрядного процессора к 16-разрядному. Но появились и новые: типы данных REAL, INTEGER, POINTER (как 20-разрядный указатель в 1 Мбайт физической памяти микроЭВМ); обработка прерываний по любому из 256 векторов прерывания; обращение к 8- и 16-разрядным портам внешних устройств (BU); управление стеком через базу сегмента (STACK-BASE) и указатель (STACKPTR).

Программы на языке PL/M-86 имеют блочную структуру (DO-блок), поэтому область действия описателей переменных и меток ограничивается этим блоком и всеми вложенными. Она прерывается, если во вложенном блоке содержится определение такого же имени. Процедура или функция определяется на любом уровне вложенности (как DO-блока, так и процедуры). Передача управления вне DO-блока или процедуры возможна лишь на уровне главной программы (DO-блок, не вложенный ни в какой другой DO-блок или процедуру). Рекурсия в языке допускается, но для этого процедура должна быть объявлена с атрибутом REENTRANT. Процедуры обработки прерываний, например, могут быть рекурсивными. Для рекурсивной процедуры или функции область локальных данных отводится динамически в стеке программы при входе и освобождается при выходе.

Имеется одно ограничение, налагаемое нашей реализацией компилятора для языка PL/M-86, — тип данных REAL не поддерживается. Это объясняется отсутствием библиотеки операций с плавающей запятой, а также недоступностью сопроцессора Intel 8087.

142284, Протвино Моск. обл., ИФВЭ, ОЭА.

ЛИТЕРАТУРА

1. Уокерли Дж. Архитектура и программирование микроЭВМ: В 2-х кн. / Пер. с англ. — М.: Мир, 1984.
2. Фаулджер Р. Программирование встроенных микропроцессоров / Под ред. Курочкина В. М. — М.: Мир, 1985.
3. Воеводин В. П. и др. Препринт. — Серпухов, 1986. — (ИФВЭ, № 86—8).
4. Ковальцов В. И., Максимов Г. М. — Препринт. — Серпухов, 1985. — (ИФВЭ, № 85—96).
5. Мобильность программного обеспечения / Под ред. Брауна П. — М.: Мир, 1980.
6. Montuelle J. CUFOM. The CERN Universal Format for Object Modules. — CERN, DD/78/21, 1978.
7. Willers I., Montuelle J. Cross Software Using a Universal Object Module Format, Proc. of the EURO IFIP 79, 1979. — P. 627—632.
8. Montuelle J. The CUFOM processors. User Manual, CERN, 1981.
9. Adams C., Willers I. CERN Intel 8080 Microprocessor Support. CERN, DD/US/37, 1978.
10. Жарков Е. А. и др. — Препринт — Серпухов, 1985. — (ИФВЭ, № 85—6).
11. MCS-86 Assembly Language Reference Manual. Intel Corporation, 1978.
12. SDK-86 MCS-86 System Design Kit. Monitor Listing. Intel Corporation, 1978.
13. MCS-86 PL/M-86 Programming Manual. Intel Corporation, 1978.
14. Gardner P. The BCPL Reference Manual. University of Essex, 1980.

Статья поступила 22.08.88

Широкое распространение дешевых микропроцессоров (МП) типа K580IK80 резко обострило проблемы разработки качественного программного обеспечения для них. Эти МП используются в интеллектуальных устройствах, контроллерах и маломощных ЭВМ, которые либо не имеют вообще, либо имеют слабые средства разработки программ. Поэтому обычно применяются кросс-системы [1], т. е. программы, работающие на более мощной, так называемой инструментальной, ЭВМ в ее операционной среде и переводящие текст на некотором языке программирования в коды МП. Полученные коды затем отлаживаются либо на самом МП, либо на инструментальной ЭВМ с помощью эмулятора кодов МП.

В настоящей работе описан опыт эксплуатации кросс-системы i85 [2]. Кросс-система обеспечивает трансляцию программ, написанных на языке Си [3] (Small-C [4]) или ассемблере, связывание их в единый загрузочный модуль и диалоговую отладку на инструментальной ЭВМ. В состав i85 входят:

Small-C — компилятор с диалекта языка Си на ассемблер МП. Поддержка времени выполнения включает целочисленную арифметику, работу со строками, форматный ввод-вывод и т. п.;

as85 — кроссассемблер, создающий переместимые коды и поддерживающий условную трансляцию, программные секции и т. д.;

pas — препроцессор для кроссассемблера, позволяющий использовать структурные управляющие операторы, такие как do, if — else, while и др.;

Id85 — редактор связей, связывающий и настраивающий по точке загрузки переместимые модули, библиотеки таких модулей и загрузочные модули;

d85 — диалоговый отладчик программ со встроенным интерпретатором кодов МП, поддерживающий чтение-запись 64 Кбайт памяти, портов ввода-вывода и регистров МП, обработку и имитацию прерываний, работу с точками останова и возможности подключения пользовательских имитаторов устройств;

ssc — программа связи частей системы.

Все компоненты i85 написаны на языке Си, функционируют в ОС ИНМОС [5] и UNIX [6] и легко могут быть перенесены в любую ОС, имеющую развитую систему программирования на Си. В частности, реализован перенос кросс-системы в МОС ВП 32-разрядной ЭВМ SM1700 [7] и MS-DOC IBM PC.

Наличие компилятора Small-C позволяет использовать для подготовки программ средства автоматизации программирования, предоставляемые инструментальной ОС. Тексты программ создаются на инструментальной ЭВМ с помощью экранных редакторов, генераторов программ [8], систем сопровождения исходных текстов и других средств. Затем они транслируются базовым Си-компилятором в коды инструментальной ЭВМ и отлаживаются на ней. Для отладки программ, связанных со специфическими особенностями аппаратуры, могут создаваться программы-имитаторы, разработка которых на языке высокого уровня не сложна. Для связи с реальной аппаратурой используются модули, написанные на ассемблере. Отлаживать эти модули можно с помощью диалогового отладчика d85. Затем программы переводятся в коды, готовые для выполнения на МП.

Для переносимости программ с инструментальной ЭВМ на МП требуется обеспечить совместимую поддержку времени выполнения программ на Small-C. В i85 имеется поддержка ввода-вывода через порты МП, совме-

стимая со стандартным вводом-выводом, однако полностью отсутствуют средства для работы с диском. В связи с этим была поставлена задача создания совместимой с ИНМОС поддержки файловой системы на внешнем устройстве прямого доступа (диске), работающей с программами на Small-C.

На языке Small-C были реализованы модули управления файловой системой, совместимые по программному интерфейсу с файловой системой ИНМОС. Модули управления файловой системой представляют диск в виде набора независимых файлов, доступных по именам, обеспечивают прямой и последовательный доступ к файлам (последовательные блоки файла могут располагаться в несмежных областях диска); для диска поддерживается общий список свободных блоков (плохие блоки отслеживаются и исключаются из работы). Эти программы связываются с внешним устройством через драйвер, представляющий диск в виде массива блоков. Размеры и количество блоков зависят от конкретного внешнего устройства. Такой драйвер был подготовлен для НГМД МФ-3200 интеллектуального терминала ТАР-34 (ЕС8534). Файловая система не зависит от конкретной аппаратуры. В частности, она работает в ИНМОС, пользуясь имитатором НГМД и его драйвером.

Совместимость программного интерфейса позволяет полностью отладить программу, работающую с диском, пользуясь средствами инструментальной ОС. Например, для ТАР-34 было написано несколько сервисных программ, обслуживающих файловую систему: универсальная программа копирования и слияния файлов, программы удаления и переименования файла, простейший экраный редактор текстов и др. Ни одна из этих программ не потребовала изменений в работе с файлами после отладки их в ИНМОС. Эксплуатация файловой системы показала ее удобство, гибкость и надежность.

Разработка файловой системы послужила экспериментальным проектом, подтвердившим работоспособность кросс-системы и позволившим сделать вывод о целесообразности ее применения. Вся работа по созданию программ управления файловой системой и ряда сервисных программ заняла менее двух человеко-месяцев. Программы управления файловой системой занимают 1400 строк на Small-C (12 Кбайт кода МП). Общий объем программ составил около 2400 строк на Small-C, из которых приблизительно 10% — комментарии, и 120 строк на ассемблере (драйвер НГМД). Размеры эмулятора НГМД — 170 строк, сервисных программ (без экранного редактора) и программ тестирования файловой системы — 800 строк.

Small-C — достаточно широкое подмножество стандартного языка Си. Поддержка времени выполнения (включая описанную файловую систему) совместима с поддержкой в ИНМОС. Поэтому сроки разработки программ с использованием кросс-системы близки к срокам разработки программ аналогичного объема и сложности в ИНМОС.

Опыт реализации и эксплуатации описанной файловой системы показал, что i85 — эффективное средство подготовки программ для различных устройств, собранных на МП К580ИК80. Использование i85 значительно сокращает сроки разработки программ, увеличивает их качество и облегчает сопровождение и модификацию.

Телефон 344-51-79, Москва

ЛИТЕРАТУРА

1. Слободенюк А. К., Погорелый С. Д., Вайсбанд С. Г. Кроссовая система отладки программного обеспечения микропроцессоров К580ИК80 на малых ЭВМ // УСИМ. — 1982. — № 3. — С. 38—41.
2. Стерин М. Ф. Кросс-система для разработки программ в кодах микропроцессоров К580ИК80, INTEL 8080/8085 // Сб. науч. тр. «Вопросы проектирования и диагностики вычислительных устройств и комплексов». — М.: ИНЭУМ, 1986. — С. 77—81.

3. Керниган Б., Ритчи Д. Язык программирования Си. — М.: Финансы и статистика, 1985. — 280 с.
4. James E. Hendrix. The Small-C Handbook. — Prentice-Hall, Reston, Virginia 2209, 1984. — 255 p.
5. Беляков М. И., Ливерсовский А. Ю., Семик В. П., Шяудкулис В. И. Инструментальная мобильная операционная система ИНМОС. — М.: Финансы и статистика, 1985. — 232 с.
6. Банахан М., Раттер Э. Введение в операционную систему UNIX. — М.: Радио и связь, 1986. — 343 с.
7. Столяр Л. И., Холмянский А. С., Шапошников В. А. МОС ВП — базовая операционная система ЭВМ СМ1700 // Сб. науч. тр. «Технические и программные средства высокопроизводительных комплексов СМ ЭВМ». — М.: ИНЭУМ, 1987. — С. 26—35.
8. Stephen C. Johnson. YACC — Yet Another Compiler // In: UNIX Programmer's Manual. — 1983. — Vol. 2. — Holt, Rinehart and Winston. — P. 353—388.

Статья поступила 15.03.88

УДК 681.323

Д. В. Барковский, И. Г. Креславский, Н. В. Сеницын

ТУРБОБИТ — ПРОГРЕССИВНАЯ СИСТЕМА ПОДГОТОВКИ МИКРОПРОГРАММ

Система ТУРБОБИТ предназначена для оперативной подготовки микропрограмм на этапе отладки микропроцессорных устройств, выполненных на базе секционированных микропроцессорных комплектов (МПК). ТУРБОБИТ — часть персональной системы проектирования, использующей в качестве инструментальной ЭВМ микроЭВМ «Электроника 60», ДВК2М, ДВК3.

Основные особенности ТУРБОБИТ:

поддержка однопрограммной системой различных этапов проектирования микропрограмм [1, 2],

хранение функциональных программных модулей и исходных текстов микропрограмм в оперативной памяти (ОП) инструментальной ЭВМ,

размещение микрокода в памяти эмулятора [3],

высокая оперативность проектирования микропрограмм,

совместимость с микропрограммным ассемблером МАСС

[4] и системой микропрограммирования МЕТАБИТ.

Система реализована методом погружения в ОС инструментальной ЭВМ. Общая схема ее размещения в ОП приведена на рисунке. Она работает в диалоге с оператором и реализует следующие функции:

ввод текста рабочей микропрограммы;

экранное редактирование текста, настройку на систему команд микропроцессорного устройства, ассемблирование микропрограммы;

детектирование, идентификацию ошибок в микропрограммах с автоматическим переходом в режим их редактирования;

поиск по заданному адресу соответствующей микропрограммной строки и переход к ее редактированию;

вывод сокращенного справочника рабочего или системного дисков инструментальной ЭВМ;

запись текста микропрограммы на диск;

тестирование и контроль быстродействующей памяти эмулятора;

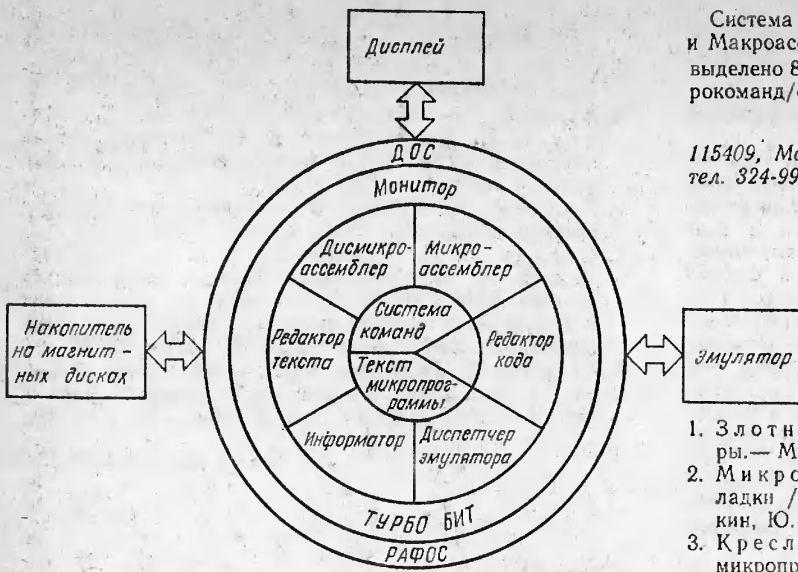
транзитную загрузку эмулятора микрокодом и сохранение образа его памяти на диске.

При ассемблировании генерируемый код может передаваться в эмулятор по адресу, указанным в микропрограмме, записываться на диск в стандартном формате либо выводиться на дисплей для визуального контроля.

В режиме оверлирования модулей ТУРБОБИТ позволяет редактировать микропрограммную память эмулятора и дисассемблировать микрокоманды.

Система ТУРБОБИТ реализована на языках Паскаль и Макроассемблер. Для хранения текста микропрограммы выделено 8 Кбайт ОП. Скорость трансляции — 10...20 микрокоманд/с.

115409, Москва, Каширское ш., 31, МИФИ, кафедра 26; тел. 324-99-81



ЛИТЕРАТУРА

1. Злотник Е. М. Секционированные микропроцессоры. — Минск: Наука и техника, 1984.
2. Микропроцессоры: системы проектирования и отладки / В. А. Мясников, М. Б. Игнатъев, А. А. Кочкин, Ю. Е. Шейнин. — М.: Энергоатомиздат, 1985.
3. Креславский И. Г., Жарков А. П. Эмулятор микропрограммной памяти // Микропроцессорные средства и системы. — 1989. — № 4.
4. Креславский И. Г., Барковский Д. В., Жарков А. П., Синицын Н. В. Микропрограммный ассемблер МАСС // Микропроцессорные средства и системы. — 1989. — № 4.

Структурная схема системы ТУРБОБИТ

УДК 681.323

И. Г. Креславский, Д. В. Барковский

Сообщение поступило 11.02.88

МЕТАБИТ — УНИВЕРСАЛЬНАЯ СИСТЕМА МИКРОПРОГРАММИРОВАНИЯ

Система микропрограммирования (СМ) МЕТАБИТ предназначена для разработки ПО специализированных вычислительных и управляющих устройств, выполненных на базе секционированных микропроцессорных комплектов (МПК).

В ее состав входят перемещаемый микропрограммный ассемблер (ПМА),

компоновщик и библиотечарь. Особенности системы:

реализация на персональных и микроЭВМ в среде дисковых ОС, поддержка модульного принципа проектирования микропрограмм; значительная мощность микропрограммного ассемблера, использование надежных механизмов кодирования микрокоманд.

Модульный принцип проектирования позволяет: применять структурные методы проектирования ПО, проводить раздельную трансляцию модулей микропрограммы, объединять стандартные модули в библиотеки, сокращать время трансляции микропрограмм, увеличивать ресурс ОП при трансляции.

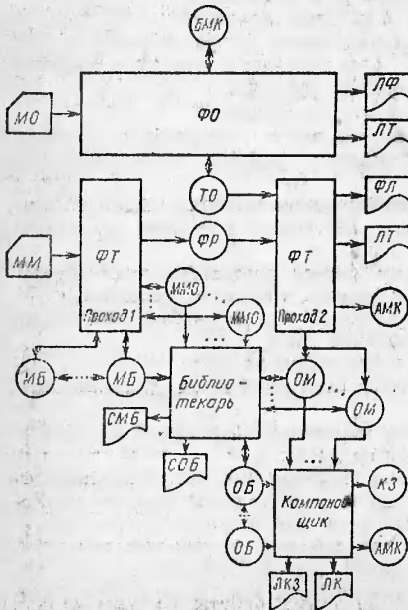
Тестирование СМ МЕТАБИТ показало, что по техническим характери-

стикам она не уступает, а по ряду функциональных возможностей превосходит аналогичные системы ведущих зарубежных фирм [1—3].

Общая технологическая схема работы системы микропрограммирования представлена на рисунке. Основа системы — перемещаемый микроассемблер [4]. Работа ассемблера разделена на две фазы: фазу определений (обрабатывает описания системы микропроцессорного устройства) и фазу трансляции (ассемблирует микропрограммы и генерирует объектный или абсолютный машинный код).

Компоновщик микропрограммных объектных модулей (ОМ) выполняет функции по сбору, размещению в памяти и разрешению всех адресных ссылок между сегментами ОМ. Он размещает сегменты в микропрограммной памяти автоматически по заданной стратегии или это делается вручную. Карта загрузки отражает результаты размещения и может документироваться. По полученному размещению выполняется сборка сегментов и генерируется машинный код.

Библиотечарь системы создает объектные и макробиблиотеки из модулей, полученных в результате работы ПМА. С его помощью можно документировать каталоги, расширить библиотеки, присоединить новые модули, удалять или экстрагировать указанный модуль из объектной библио-



Система микропрограммирования МЕТАБИТ:

МО — модуль определений, БМК — библиотека метакода, ЛФ — листинг фазы, ЛТ — листинг таблиц, ТО — таблицы определений, ММ — модуль микропрограмм, ФР — файл расширения, ММО — модуль макроопределений, МБ — макробиблиотека, ФЛ — форматированный листинг, АМК — абсолютный машинный код, ОМ — объектный модуль, ОБ — объектная библиотека, СОБ — состав объектной библиотеки, СМБ — состав макробиблиотеки, КЗ — карта загрузки, ЛКЗ — листинг карты загрузки, ЛК — листинг кода

теки. Работа с библиотеками сокращает время доступа к хранящимся в них модулям, ускоряет трансляцию или компоновку.

Система МЕТАБИТ — функционально полная система автоматизации проектирования микропрограммного обеспечения. Она реализована на микроЭВМ ряда «Электроника 60» и ПЭВМ ряда ЕС1840 и доступна широкому кругу разработчиков микропроцессорной техники.

115409, Москва, Каширское ш., 31, МИФИ, кафедра 26; тел. 324-99-81

ЛИТЕРАТУРА

1. Eager M. M29 — an advanced retargetable microcode assembler // SIGMICRO Newsletter. — 1983 — № 4. — P. 92—100.
2. Wilburn D., Schleimer S. Step development tools: METASTEP language system // SIGMICRO Newsletter. — 1985. — № 4. — P. 157—164.

3. Коул Б. Инструментальная система на базе персонального компьютера для программистов разрядно-модульных машин // Электроника. — 1987. — № 14. — С. 45—48.
4. Креславский И. Г., Барковский Д. В., Жарков А. П., Синец Н. В. Микропрограммный ассемблер МАСС // Микропроцессорные средства и системы. — 1989. — № 4.

Сообщение поступило 11.02.88

УДК 681.3.06.068

М. Д. Сафир, В. М. Сиянко, М. К. Урсатий

СИСТЕМА ОТЛАДКИ ДЛЯ СЕКЦИОНИРОВАННЫХ МИКРОПРОЦЕССОРОВ

Система отладки (СО) для высокопроизводительных специализированных микропроцессорных систем (МПС) на основе микропрограммируемых секционированных БИС состоит из управляющей микроЭВМ «Электроника 60», модуля отладки и имитатора ПЗУ (рис. 1).

Управляющая микроЭВМ связана с отлаживаемой МПС через имитатор ПЗУ по шине микрокоманд (ШМК), через модуль отладки с любой из доступных внутренних шин МПС: адреса микрокоманд (ШАМК), данных (ШД), регистра состояния (ШРС) процессора и по цепи управления (УПР) с устройством синхронизации МПС для управления вычислительным процессом во времени.

Данные от управляющей микроЭВМ передаются в МПС через имитатор ПЗУ, принимаются через модуль отладки.

Имитатор ПЗУ объемом 80 Кбайт (1 К×80) служит для отладки микропрограмм на макете разрабатываемой МПС и позволяет заменить ПЗУ на этапе отладки функционально эквивалентным по разрядности, объему, быстродействию и логике управления ОЗУ. Микропрограммы вводятся в имитатор, просматриваются и изменяются оператором с помощью дисплея управляющей микроЭВМ.

Модуль отладки (рис. 2) обеспечивает сопряжение канала обмена управляющей микроЭВМ с МПС и имитатором. Связь между устройствами, подключенными к каналу обмена, осуществляется по принципу «активный-пассивный». Данные через канал обмена передаются асинхронно с помощью специальных сигналов синхронизации (СС) К ВВОД Н, К ВЫВОД Н, К СИАН, т. е. на инициализирующий сигнал обмена данных от активного устройства (управляющая ЭВМ) должен поступать ответный сигнал от выбранного пассивного устройства К СИПН.

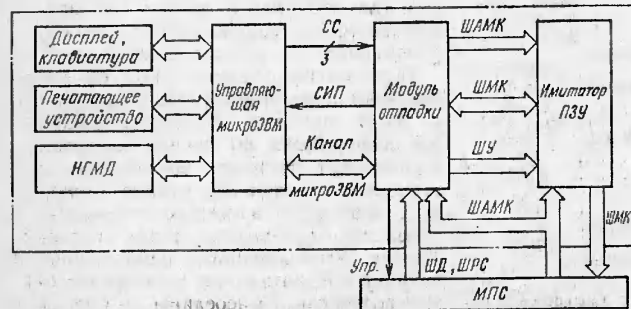


Рис. 1. Структурная схема системы отладки

СРЕДСТВА ОТЛАДКИ МП-СИСТЕМ

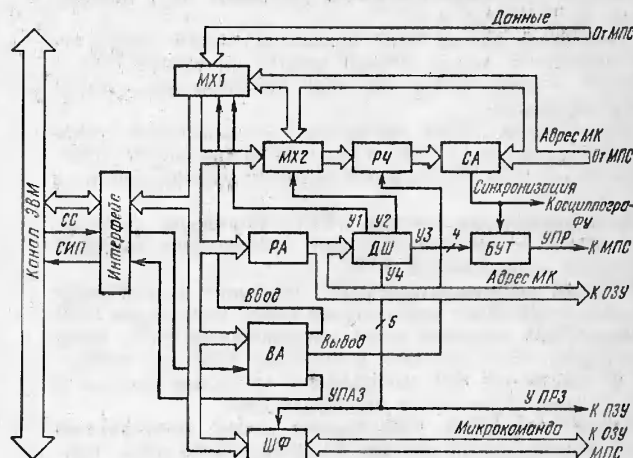


Рис. 2. Функциональная схема модуля отладки

К каналу обмена модуль отладки подключен двумя портами. Один из них с адресом 177540₈ — порт вывода. Через него передаются команды управления системой отладки. Другой с адресом 177542 — порт ввода-вывода. Он обеспечивает передачу данных с микроЭВМ в систему отладки и получение запрашиваемой информации.

Один цикл работы системы отладки приходится на минимум два цикла работы микроЭВМ (рис. 3). Между передачей команды и данных микроЭВМ может выполняться работа по подготовке данных. Старшие четыре бита команды отведены под код команды, а остальные представляют

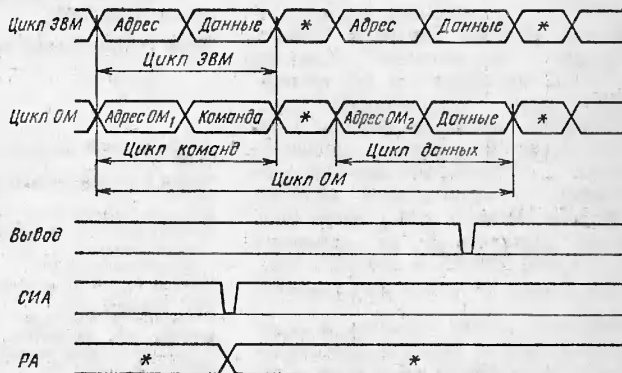


Рис. 3. Временная диаграмма работы системы отладки

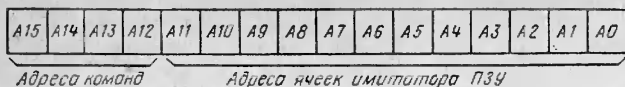


Рис. 4. Структура команды системы отладки

собой адреса ячеек памяти имитатора, в которые необходимо занести информацию (рис. 4).

Интерфейс (см. рис. 2) обеспечивает связь между каналом микроЭВМ и узлами отладочного модуля, подключен к внешнему разьему канала микроЭВМ.

Мультиплексор адреса-данных (МХ1) используется для выдачи в канал микроЭВМ информации о состоянии регистров отлаживаемой МПС. На входы МХ1 подключены: ШД, ШАМК и РС.

Выделитель адреса (ВА) предназначен для записи кода команды и адреса ячейки памяти имитатора ПЗУ в регистр адреса (РА), где они хранятся до следующего обращения.

Дешифратор (ДШ) организует параллельный обмен информацией между МПС и микроЭВМ, формирует управляющие сигналы У1...У4 и обеспечивает дешифрацию кода команды.

Блок управления тактами (БУТ) формирует сигналы, разрешающие или запрещающие прохождение тактовой частоты в отлаживаемой МПС.

Шинный формирователь (ШФ) позволяет организовать параллельный обмен информацией между имитатором ПЗУ и микроЭВМ, передать адрес микрокоманды МПС, которая должна быть занесена в имитатор ПЗУ или выведена из имитатора для отображения на экране дисплея с последующей записью на магнитный диск.

Канал микроЭВМ представляет собой 16-разрядную шину адреса-данных, поэтому 8-, 10-разрядное слово передается пятью циклами по 16 бит в каждом. ДШ формирует на шине управления (ШУ) сигналы поочередного подключения канала микроЭВМ к определенной области памяти имитатора ПЗУ. С помощью ШД организуется двусторонняя магистраль обмена информацией. Направление обмена данными выбирается сигналами на ШУ.

Мультиплексор адреса останова программы (МХ2) предназначен для занесения адреса останова из канала

микроЭВМ или с шины адреса микрокоманды в регистр числа (РЧ). Адрес поступает в узел сравнения адресов (СА) для сравнения с последовательно меняющимися адресами микрокоманд МПС. При их совпадении формируется сигнал для БУТ, управляющий работой устройства синхронизации отлаживаемой МПС.

Система отладки обеспечивает диалог с пользователем; ввод, хранение и редактирование микропрограммы отлаживаемой МПС; сбор, хранение, индикацию и документирование информации; управление ходом вычислительного процесса в разрабатываемой МПС, в том числе установку начальных адресов микропрограммы пользователя, организацию пошагового режима, исполнение микропрограммы до выполнения заданного условия; управление исполнением микропрограммы отлаживаемой МПС в реальном масштабе времени при любом быстродействии.

Управляющая программа и микропрограммы для отлаживаемой МПС хранятся на диске. Управляющая программа состоит из подпрограмм, отвечающих за определенный режим работы системы:

ввод с клавиатуры дисплея любой двоичной информации (в частности, микропрограммы для отлаживаемой МПС);

запись введенной с клавиатуры информации на магнитный диск;

коррекцию введенной информации;

перезапись массива данных из любой области ОЗУ микроЭВМ «Электроника 60» в любую область памяти имитатора ПЗУ;

чтение массива данных из любой области имитатора ПЗУ с отображением на экране дисплея;

пошаговое выполнение микропрограммы МПС;

задание режима исполнения микропрограммы с любого адреса;

задание точки останова исполняемой микропрограммы с отображением на экране дисплея состояния шин, доступных управляющей, программе в момент останова;

остановку выполнения микропрограммы;

запрос текущего состояния шин отлаживаемой МПС, когда микропрограмма остановлена.

Телефон 271-41-23, Киев

Статья поступила 7.01.88

УДК 681.326.7

С. Я. Иванцов, В. Я. Клим, В. И. Мамай, Н. В. Посупонько, Ю. В. Терпугов

АВТОМАТИЗИРОВАННАЯ КОНТРОЛИРУЮЩАЯ СИСТЕМА ПАКС-МК

Особое место в ряду устройств контроля цифровых узлов РЭА занимают компактные персональные автоматизированные системы контроля индивидуального пользования. Система ПАКС МК предназначена для автоматизированного функционально-параметрического контроля сложных цифровых узлов в реальном масштабе времени с использованием методов тестового и сигнатурного анализа. Объектами контроля (ОК) могут быть изделия, выполненные на элементах ТТЛ, ТТЛШ, КМОП и других, совместимых с ними по логическим уровням.

ПАКС-МК может работать в автономном режиме в качестве персональной системы контроля и в составе мощного информационно-вычислительного комплекса в качестве удаленного терминала, подключенного к приборному

интерфейсу типа МЭК-магистраль.

Основные технические характеристики ПАКС-МК

Число контролируемых каналов	1...61 (при объеме единички четырех приборов — до 244)
Длина тестовой последовательности, такт	8000
Число повторов любого фрагмента теста	2 ¹⁶ —1
Скорость подачи теста, такт/с	60...4·10 ⁶
Уровень Лог. 1, В (при I _{вых} ≤ 30 мА):	
фиксированный	4,5
программируемый	3...12
Уровень Лог. 0, В, не более (при I _{вых} ≤ 30 мА)	0,5
Погрешность измерения логических уровней, мВ, не более	±50
Напряжение источников питания объекта контроля, В:	
фиксированного	5±2 %
программируемого	(3...12)±2%
Входное сопротивление канала, кОм, не менее	30

Система содержит прибор, выполненный в конструктиве НАДЕЛ-75А, и стандартный дисплей. ОК подключается к прибору через разъем типа ГРПМШ1-61. Выводы ОК соединены с расположенными на передней панели прибора гнездами, с помощью которых осуществляется подключение к ОК дополнительных источников питания, измерительных приборов и т. п.

При объединении нескольких приборов (до четырех) в стойку ОК подключается с помощью гибких кабелей, распаянных на разъем пользователя.

Тестовая информация (ТИ) об ОК хранится на квазидиске, установленном в нише прибора. Подготавливается эта информация до начала контроля и включает тестовые воздействия и эталонные отклики ОК, данные о входных, выходных и незадействованных контактах, фрагментах теста, подлежащих закликиванию, напряжениях питания ОК, логических уровнях тестовых воздействий и скорости их подачи на ОК.

При тестировании ОК оператор под-

ключает к ПАКС-МК проверяемую схему и дает команду о начале контроля. При этом система автоматически настраивается на проверку данного ОК: тестовые воздействия и эталонные отклики переписываются с квазидиска в буферную канальную память, устанавливается скорость подачи теста, напряжение питания ОК в соответствии со служебной информацией, также хранящейся на квазидиске. Затем система выдает тестовые наборы с заданной скоростью на входы ОК, считывает с его выходов отклики и сравнивает их с эталонами, хранящимися на квазидиске. При обнаружении несовпадений информация об ошибках выводится на экран дисплея.

Для контроля схем, содержащих многоразрядные счетчики или регистры с обратными связями, требующих подачи на вход серий длинных (до 10^6 тактов) последовательностей, предусмотрена возможность закидывания на рабочей частоте любых фрагментов тестов. Каналы контроля в процессе проверки могут переключаться с выдачи теста на прием откликов и наоборот. При этом скорость подачи теста не уменьшается.

Для диагностики ОК и локализации неисправности используется шуп, с помощью которого с любой точки ОК можно считать отклик и вывести соответствующую временную диаграмму на экран, снять сигнатуру и измерить напряжение логических уровней. Для локализации неисправности предусмотрена возможность работы с осциллографом. При этом на ОК циклически подаются входные воздействия (либо их фрагменты), а на экране осциллографа наблюдаются сигналы в различных точках ОК.

В случае короткого замыкания во входных цепях ОК выходы прибора автоматически отключаются и на дисплее индицируется номер неисправного канала.

При отключении питания системы ТИ сохраняется на квазидиске емкость 64 Кбайт, построенном на основе РПЗУ с электрическим или ультрафиолетовым стиранием либо КМОП ОЗУ с резервным питанием. На этапе отладки теста целесообразно использовать квазидиск, содержащий микросхемы КМОП ОЗУ КР537РУ10 либо РПЗУ с электрическим репрограммированием КР1609РР1, КР558РР2, КР573РР2. Отлаженные тесты удобно хранить на квазидиске, в который установлены ИМС РПЗУ с ультрафиолетовым стиранием К573РФ5. Для занесения ТИ предусмотрен специальный режим. Долговременно хранить ТИ можно на бытовом кассетном магнитофоне, сопряженном с которым обеспечено в системе.

Во всех режимах, кроме тестирования ОК, ПАКС-МК работает под управлением контроллера, построенного на основе универсального процессора К580ВМ80А. Режимом тестирования управляет спецпроцессор (рис. 1).

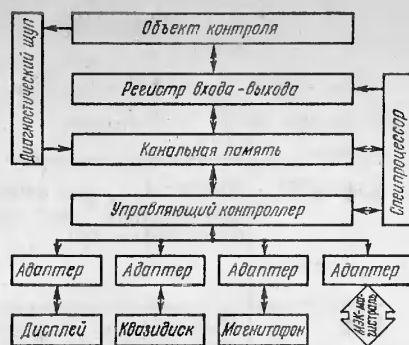


Рис. 1. Структурная схема ПАКС-МК

Управляющий контроллер с периферийными устройствами выполнен на основе МПК серии К580 [5]. Канальная память (КП) представляет собой ОЗУ емкостью 64 Кбайт с различной организацией по отношению к управляющему контроллеру и спецпроцессору: 65536 8-разрядных слов и 8096 64-разрядных слов соответственно, что позволяет достичь высокой скорости работы системы в режиме тестирования ОК и обеспечивает удобный доступ оператора к ТИ.

Спецпроцессор (СП) по возложенным на него функциям аналогичен контроллеру прямого доступа к канальной памяти и представляет собой устройство, ориентированное на выполнение четырех операций:

СЧИТЫВАНИЕ-МОДИФИКАЦИЯ-ЗАПИСЬ ОДНОГО СЛОВА КП, что соответствует подаче на входы ОК одного такта теста и записи его отклика;

МОДИФИКАЦИЯ РЕГИСТРА ВХОДА-ВЫХОДА — переводу необходимых каналов из режима выдачи теста в режим считывания откликов и наоборот;

ЦИКЛ — закидыванию фрагмента теста;

СТОП — окончанию работы СП и переводу его в исходное состояние.

СП оперирует 64-разрядными словами, хранящимися в КП. Формат команды СП представлен на рис. 2. В разрядах 1..61 располагается поле данных, над которыми производится действие в соответствии с кодом операции, хранящимся в 62-м и 63-м разрядах; 64-й разряд — служебный, в нем фик-

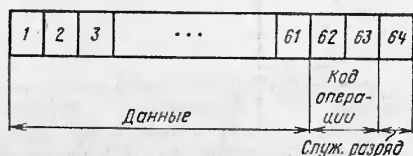


Рис. 2. Формат команды спецпроцессора

сируются ошибки типа «промежуточные уровни», возникающие в любом канале при считывании откликов ОК. Программу работы СП (т. е. информацию в 62-й и 63-й разряды КП) готовит управляющий контроллер, записывая по нужному адресу код требуемой операции.

Программное обеспечение ПАКС-МК (монитор ПАКС-МК) предназначено для управления системой в различных режимах работы, подготовки ТИ и файловой организации, хранения ее на квазидиске. Монитор построен по модульному принципу и работает в диалоговом режиме. Возможен вывод краткой инструкции. Предусмотрена защита от ошибочных действий оператора. Монитор состоит из подсистем начального диалога и диспетчера, подготовки тестовой информации, тестирования объекта контроля, управления файлами на квазидиске или НМЛ.

В подсистеме начального диалога и диспетчеризации оператору предлагается меню возможных режимов работы. В случае необходимости краткого знакомства с режимами можно обратиться к справке.

Подсистема подготовки тестовой информации включает три отдельных функциональных модуля. Первый модуль — экранный редактор ТИ — позволяет вводить с клавиатуры и корректировать тесты, представленные в виде временных диаграмм. Здесь же набирается служебная информация о распределении входов-выходов, возможных циклах в тестовой диаграмме и т. д. Второй модуль используется для быстрого набора тестов в виде сверток (знакоперемен). Третий — позволяет отображать на экране дисплея временные диаграммы эталонных и реальных откликов ОК и используется для анализа их несовпадений, выявленных в процессе тестирования ОК.

Подсистема тестирования ОК содержит модули, реализующие взаимодействие системы с ОК и анализ откликов ОК. Эта подсистема обеспечивает программирование СП, запуск и аварийный останов, а также обслуживание портов, участвующих в процессе тестирования ОК.

Подсистема управления файлами задает файловую организацию хранения ТИ на квазидиске, реализует просмотр каталога, а также запись (чтение) и исключение файлов.

ПАКС-МК может найти применение при контроле функционирования схем в процессе их изготовления в условиях серийного производства, контроле параметров схем при проведении испытаний, проверке макетов в лабораторных условиях, выходном контроле БИС и т. д.

Телефон 65-45-81, Ростов-на-Дону

Статья поступила 30.03.88

В. В. Балюк, С. П. Кобзар

ДИАГНОСТИРОВАНИЕ УСТРОЙСТВ НА БАЗЕ МИКРОПРОЦЕССОРА КР580ИК80А

С увеличением степени интеграции микросхем диагностирование электронных приборов на их основе становится центром внимания разработчиков. Классический подход к созданию проверяющих тестов для цифровых схем [1] не может однозначно переноситься на диагностирование устройств на основе микропроцессоров (МП). Большой объем логики (порядка сотен и тысяч вентилей) сочетается в МП с ограниченным доступом к их внутреннему содержанию. Так, для полного диагностирования МП 18080, имеющего 8-битовую шину данных и систему из 76 команд, требуется создать проверяющий тест из более чем 10^{163} кодовых комбинаций [2].

Другой подход к диагностированию МП и устройств на их основе заключается в использовании генераторов псевдослучайных тестов [3]. Однако при этом [4]: а) в составе системы диагностирования требуется иметь безотказно функционирующий эталонный МП; б) трудно достичь точного моделирования процесса выполнения команд; в) для достоверного диагностирования МП требуется тест большой длины.

В работах [4] и [5] рассмотрена общая модель возникновения ошибок при работе МП, сформулированы основные требования к функциональному тесту для диагностирования МП МС68000. Эта модель организована для проверки МП КР580ИК80А, и здесь

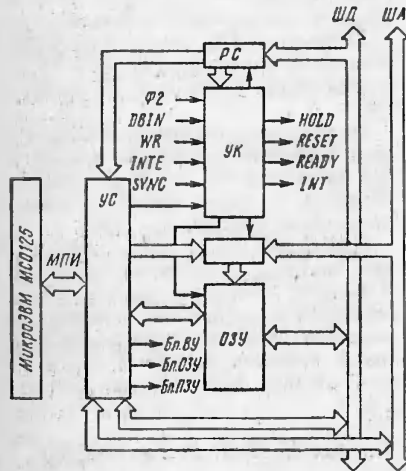


Рис. 1. Структурная схема системы контроля МП КР580ИК80А:

УС — устройство сопряжения; УК — устройство управления каналом; СФА — схема формирования адреса; РС — регистр состояния

Команда	Ошибка	Функциональное описание теста
LXI, SPHL	Искажается адрес загрузки, одновременно выбирается более одного ПИ	РГ загружаются различающимися кодами, проверяется содержимое РГ и флажков, РГ загружаются инверсными кодами в обратном порядке, проверяется содержимое РГ и флажков
POP	Неправильно адресуется ОЗУ, содержимое СТ загружается не в те РГ, одновременно выбирается более одного ПИ	SP устанавливается на первый СТ, из первого СТ загружаются все РГ (включая РГ F) различающимися кодами. SP устанавливается на второй СТ, во второй СТ записывается содержимое всех РГ (включая РГ F), SP устанавливается на третий СТ, из третьего СТ загружаются все РГ инверсными кодами в обратном порядке, SP устанавливается на четвертый СТ, в четвертый СТ записывается содержимое всех РГ
PUSH	Неправильно адресуется ОЗУ, содержимое не тех РГ загружается в СТ, одновременно выбирается более одного ИИ	SP устанавливается на третий СТ, из третьего СТ загружаются все РГ инверсными кодами в обратном порядке, SP устанавливается на четвертый СТ, в четвертый СТ записывается содержимое всех РГ
LDA, STA	Неправильно выполняется обмен между РГА и ШД, одновременно выбирается более одного ИИ или ПИ, неправильно адресуется ОЗУ	Проверяются совместно с PUSH, POP, LXI SP. После загрузки РГ (включая РГ F) различающимися кодами в РГА загружается по LDA отличный от них код и записывается по STA в ОЗУ СК по другому адресу. Затем по LDA в РГА загружается инверсный код, который записывается в другое место ОЗУ СК или STA. Далее в СТ загружается содержимое всех РГ для последующего контроля
DI, EI	Неправильно установлено состояние триггера разрешения прерываний, не выполняется запрещение или разрешение прерываний	Проверяются совместно с одной из команд RST. Выполняются команды DI и HLT, анализируется состояние выхода разрешения прерываний МП, выставляется запрос прерывания и проверяется невыполнение ПП обработки запроса прерывания. Выполняются команды EI и HLT, анализируется состояние выхода разрешения прерываний МП, выставляется запрос прерывания и проверяется правильность выполнения ПП обработки прерывания и возврата из нее
RST	Искажается принимаемый байт, неправильно формируется адрес и выполняется возврат из прерывания	Дважды выполняется цепочка всех рестартов при различных значениях флажков, проверяется последовательность выполнения рестартов, содержимое флажков и СТ. Команды RST проверяются после проверки команды CMR
Переходы	Искажается адрес перехода и содержимое РГ, неправильная реакция на условия перехода	Загружаются из СТ регистры, проверяется выполнение команд переходов при различных значениях флажков (РГ флажков загружается из СТ); значения РГ и флажков записываются в стек для последующего анализа. При ошибочном выполнении перехода МП переходит в состояние останова; при правильном — происходит обход команды HLT
CALL	Неправильно передается управление и выполняются операции с СТ; искажается содержимое РГ	Команды проверяются аналогично командам переходов с фиксацией факта входа в подпрограмму и выхода из нее путем записи определенных кодов в фиксированные ячейки ОЗУ СК

Команда	Ошибка	Функциональное описание теста
MVI, MOV, XCHG, LDAX, STAX	Нет призна информации, искажается выбираемый байт, неправильно выбирается ИИ или ПИ, одновременно выбирается более одного ИИ или ПИ, искажается адрес	Команды MVI проверяются совместно с CPI и CMP путем загрузки в РГ различных значений, сравнения, анализа значений флажков и проверки сохранности содержимого РГ; команда XCHG — после MVI и CMP аналогично MVI; команды MOV — после INR и DCR последовательной пересылкой значений из каждого регистра во все другие РГ с выполнением инкремента-декремента для различимости и проверки содержимого РГ и флажков; команды LDAX и STAX — после INX, DCX путем чтения и записи в смежные ячейки ОЗУ. При отработке команд устанавливается сохранность РГ и флажков
CPI, CMP	Искажается выбираемый байт, неправильно выбирается ИИ или ПИ, неправильно выставляются флажки, результат записывается в РГА, искажается адрес	
LHLD, SHLD	Искажается адрес, отсутствует прием байта, искажается выбираемый байт, неправильно выбирается ИИ или ПИ, одновременно выбирается более одного ИИ или ПИ, отсутствует приращение адреса	Чтение и запись в ОЗУ с контролем сохранности РГ и флажков и проверки записанных значений
INRr, DCRr	Отсутствует прием байта, искажается выбираемый байт, неправильно выбирается ИИ или ПИ, одновременно выбирается более одного ИИ или ПИ, неправильно выставляются флажки, искажается адрес	Заполнение всех РГ кодом 0, последовательное увеличение и уменьшение содержимого РГ с проверкой на каждом этапе сохранения неизменяемыми РГ и правильности установки флажков
INR M, DCR M		Изменение кода в адресуемой ячейке ОЗУ, установка флажков и сохранность всех РГ
INX, DCX		Изменение кода в регистровой паре, сохранность флажков и содержимого РГ
NOP	Искажается содержимое РГ	Сохранность флажков и содержимого РГ
XTHL	Искажается информация при обмене между СТ и РГ, неправильно выбирается ИИ или ПИ, одновременно выбирается более одного ИИ или ПИ, неправильно выполняются операции с СТ	Правильность обмена между верхушкой стека и регистровой парой HL при различном содержимом СТ, РГ, флажков
Арифметическая и логическая	Искажается выбираемый байт, неправильно выбирается ИИ или ПИ, одновременно выбирается более одного ИИ или ПИ, неправильно выполняются операции и выставляются флажки	Правильность результата выполнения операции, установки флажков и сохранность РГ при различных комбинациях содержимого РГ и непосредственных операндов
PCHL	Искажаются выбираемые байты, неправильно выбирается ИИ или ПИ, одновременно выбирается более одного ИИ или ПИ	Запись в регистровую пару HL адреса метки, занесение в счетчик команд содержимого регистровой пары HL, изменение после перехода содержимого определенных ячеек ОЗУ, сохранность РГ и флажков

представлены практические рекомендации по ее применению.

Предположим, что возможны следующие ошибки при выполнении произвольной команды I_i: а) команда не выполняется; б) вместо команды I_i выполняется команда I_j; в) одновременно выполняются команды I_i и I_j [5]. Кроме того, проверяющие тесты разработаны с учетом того, что при выполнении команды I_i может неправильно выбираться источник или приемник информации, искажаться флажки и информация при выборе из источника и передаче в приемник, одновременно выбирается более одного источника или приемника информации.

Анализ структуры и системы команд МП показал, что часть функций либо нельзя проверить только программным путем, например правильность отработки команд HLT, DI, EI, NOP, IN, OUT, либо для этого требуются проверяющие тесты большой длины.

Разработанная система контроля (СК), структура которой представлена на рис. 1, позволяет увеличить полноту диагностирования МП и уменьшить длину тестов.

МикроЭВМ MC0125 выбрана в качестве управляющей машины благодаря высокому быстродействию, широкому набору периферийных устройств, простоте сопряжения с внешними устройствами, развитому ПО и наличию семейства машин, программно совместимых с ней. УС может быть реализовано на основе интерфейса пользователя И5 или устройства параллельного обмена И2, входящего в состав MC0125. УК формирует сигналы сброса, захвата шин, запроса прерывания, устанавливает сигнал READY при переводе МП в состояние ожидания, организует взаимодействие МП и микроЭВМ с ОЗУ СК, управляет записью в РС и имитирует обращение к ВУ МП. СФА обеспечивает доступ к ОЗУ СК со стороны микроЭВМ (при этом МП переводится в состояние «Захват») и МП. Если в составе проверяемой микропроцессорной платы есть собственное ОЗУ, то его необходимо блокировать подачей сигнала «Бл.ОЗУ», вырабатываемого под управлением MC0125.

Стратегия проверки заключается в последовательном наращивании объема проверенных команд МП. В качестве проверяющих тестов применяются законченные программные модули, написанные в системе команд МП КР580ИК80А и позволяющие локализовать с точностью до команды (цепочки команд) сбой в работе. Окончание очередного теста фиксируется по результату отработки МП команды HLT, а правильность отработки проверяемых команд определяется микроЭВМ MC0125 сравнением с эталонным результатом, полученных после выполнения очередного теста. Сохранение регистров, установка и сброс флажков

Команда	Ошибка	Функциональное описание текста
Вызов ПП и возврат из нее	Искажается адрес перехода, неправильная реакция на условия перехода, неправильно выполняются операции с СТ, искажается содержимое РГ	Правильность выполнения или невыполнения вызова ПП и возврата из нее при различных значениях флажков, сохранность РГ и флажков
IN, OUT	Искажаются адрес, выбираемый байт и содержимое РГ	Имитация обращения к внешнему устройству обращением к ОЗУ СК. Выполняется команда OUT по всем возможным адресам, затем МП сравнивает содержимое имитируемых ячеек ОЗУ СК с эталоном. Далее при помощи команды IN читается содержимое ячеек, заполненных командой OUT, и МП сравнивает их содержимое с эталоном. Проверяется сохранность флажков и регистров

Примечания: УС — указатель стека, СТ — стек, РГ — регистр, РГА — аккумулятор, ПИ — приемник информации, ИИ — источник информации, ПП — подпрограмма.

проверяются посредством операций со стеком; для всех команд устанавливается правильность дешифрации кода операции; для команд типа MOV, MVI, LXI, LDA, STA, LHL, XCHG, XTHL, SPHL, PUSH, POP (кроме POP PSW), DAD, INX, DCX, IN, OUT, EI, DI, NOP, команд переходов, вызовов подпрограмм — сохранность флажков.

Проверка МП начинается с определения правильности отработки команды HLT. Если она отработана правильно, то в PC СК фиксируется сигнал HLTA, поступающий на вход требования прерывания УС и вызывающий переход микроЭВМ к программе обработки прерывания. Затем проверяется прохождение сигнала RESET и правильность работы счетчика команд МП. Для этого на шину данных (ШД), принудительно выставляется код 00 (команда NOP), МП выбирает следующую команду, ко-

торая всегда интерпретируется как NOP. МП вынужден считать NOP из каждой ячейки памяти, поэтому на шине адреса (ША) формируются все возможные двоичные коды, в чем можно убедиться с помощью осциллографа.

В таблице приведены тесты, применяемые для проверки правильности выполнения различных типов команд, и представлены классы обнаруживаемых ошибок. Программа диагностирования МП состоит из 12 тест-программ, выполнение каждой из которых заканчивается командой HLT.

В тестах 1...4 проверяется отработка останова после одиночных команд ввода-вывода и чтения-записи, в тесте 5 — обмен со стеком, в тесте 6 — работа с прерываниями, в тесте 7 — передача управления, в тестах 8...10 — арифметические и логические операции, а также пересылки и сравнения, в тесте 11 — рестарты, в тесте 12 — команды ввода-вывода. Полный дамп памяти теста 7 в HEX-формате приведен на рис. 2. После загрузки очередного теста и снятия сигналов HOLD и RESET МП выполняет его, начиная с нулевого адреса. Число кодов ошибок, которые выдаются в процессе диагностирования МП, равно 2879, число команд в исходных текстах программных модулей — 4150 (7250 байт в объектных модулях программ).

Структура программы позволяет избежать необратимых последствий при неправильной дешифрации кода операции как изменяющего, так и не изменяющего длину команды. Если изменение длины команды произошло, то оно компенсируется подбором второго

и третьего байтов команды или подбором следующей команды, не приводящим к нарушению логики функционирования тестов. Например, в трехбайтовой команде LXID при первом выполнении используется 16-ричный операнд 5859. Если ошибочная длина команды LXID равна единице, то вслед за ней будут выполнены однобайтовые команды MOV E, C и MOV E, B (коды 59 и 58 соответственно), если она равна двум, — команда MOV E, B. Другой пример: при проверке однобайтовой команды SPHL за ней следуют однобайтовые команды POP B и MOV A, A, причем последняя лишь компенсирует возможное изменение длины команды SPHL. Если длина команды SPHL равна трем, все три команды сливаются в одну, а если двум, то вслед за SPHL выполняется «балластная» команда MOVA, A.

Тесты последовательно загружаются в ОЗУ СК для выполнения, причем область ОЗУ СК, не занятая очередным тестом, заполняется командами HLT (код 76). При отсутствии ошибок тест обрабатывается до конца, формируя в определенных (контрольных) ячейках ОЗУ СК коды, позволяющие судить об успешности завершения теста.

При возникновении ошибок возможны две ситуации: тест выполняется до конца с занесением в контрольные ячейки искаженных значений; работа теста прекращается до заполнения контрольных ячеек. Однако в обоих случаях завершение теста фиксируется после отработки команды HLT.

Тесты разработаны и отлажены на мини-ЭВМ VT-20. СК применяется для диагностирования плат, содержащих МП KP5801K80A, ОЗУ K541 и ПЗУ K556.

Телефон 556-04-20, Киев

ЛИТЕРАТУРА

1. Киселев В. В., Кон Е. Л., Шеховцев О. И. Автоматизация поиска дефектов в цифровых устройствах. — Л.: Энергоатомиздат, 1986. — 96 с.
2. Уильямс Г. Б. Отладка микропроцессорных систем. — М.: Энергоатомиздат, 1988. — 253 с.
3. Fedi X., David R. Some experimental results from random testing of microprocessors. — IEEE Trans. Instrum. Meas. — 1986. — V. IM-35, N 1. — P. 78—86.
4. Brahme D., Abraham J. A. Functional testing of microprocessor. — IEEE Trans. Comput. — 1984. — V. C-33, N 6. — P. 475—485.
5. Thatte S. M., Abraham J. A. Test generation of microprocessor. — IEEE Trans. Comput. — 1980. — V. C-29, N 6. — P. 429—441.

Статья поступила 21.04.88

Рис. 2. Дамп памяти проверяющего теста 7

```

0000 31 00 01 F1 C1 D1 E1 31 65 0C C3 00 00 F5 DA 11
0010 00 F5 E9 15 00 F5 CA 19 00 F5 FA 10 00 F5 D2 21
0020 00 F5 E2 25 00 F5 C2 29 00 F5 F2 20 00 F5 D0 31
0030 00 F5 31 B4 01 C9 31 4F 00 F5 C5 D5 E5 32 23 0C
0040 31 C9 01 F1 C1 D1 E1 31 47 0C C3 40 00 F5 DA 51
0050 00 F5 EA 55 00 F5 CA 59 00 F5 FA 50 00 F5 D2 41
0060 00 F5 E2 65 00 F5 C2 69 00 F5 F2 60 00 F5 D0 71
0070 00 F5 31 D6 01 C9 31 31 00 F5 C5 D5 E5 32 27 0C
0080 31 00 01 F1 C3 08 00 76 32 26 0C 31 00 00 C0 57
0090 01 32 25 0C 31 0E 01 F1 D2 9E 04 DA 9F 00 76 32
00A0 24 0C EA 9E 00 32 23 0C CA 9E 00 32 22 0C FA 9E
00B0 00 32 21 0C F1 E2 0B 09 EA 0C 00 76 32 20 0C DA
00C0 0B 00 32 1F 0C CA 0B 09 32 1E 0C FA 0B 00 32 1D
00D0 0C F1 C2 D5 09 CA D7 09 76 32 1D 0C DA D9 00 32
00E0 1B 0C EA D0 00 32 1A 0C FA 0B 00 32 1F 0C F1 F2
00F0 F5 00 FA F6 00 76 32 10 0C DA F5 00 32 17 0C EA
0100 F5 00 32 16 0C CA F5 00 32 15 0C F1 DA F5 00 EA
0110 F5 00 CA F5 09 FA F5 00 32 14 0C F1 C9 01 F1 C3
0120 23 01 76 32 13 0C 31 00 0C C0 57 01 32 11 0C 31
0130 C0 01 F1 DA 39 01 D2 3A 01 76 32 10 0C E2 39 01
0140 32 0F 0C C2 39 01 32 0C 0C F2 39 01 32 00 0C F1
0150 EA 36 01 E2 57 01 76 32 0C E2 D2 56 01 32 0B 0C
0160 C2 56 01 32 0A 0C F2 56 01 32 09 0C F1 C9 73 01
0170 C2 74 01 76 32 0B 0C D2 73 01 32 07 0C E2 73 01
0180 32 06 0C F2 73 01 32 05 0C F1 FA 90 01 F2 91 01
0190 76 32 04 0C D2 90 01 32 03 0C E2 90 01 32 02 0C
01A0 C2 90 01 32 01 0C F1 D2 90 01 E2 90 01 C2 90 01
01B0 F2 90 01 32 00 0C 76 32 12 C9 76 37 AA 03 0A
01C0 06 AA 02 0A 02 0A 12 0A 02 55 16 50 D3 53 97 53
01D0 57 53 C7 55 36 00 76 00
    
```

СИСТЕМА ПОДГОТОВКИ ПРОГРАММ ДЛЯ ПЭВМ

Для ускорения разработки программного обеспечения ПЭВМ целесообразно автоматизировать перевод пакетов прикладных программ (ППП) с языка Фортран [1—3] ЕС ЭВМ на язык БЕЙСИК.

В основу разработанной инструментальной системы ПАКТ (пакет автоматизированного кодирования текстов) положен сравнительный анализ операторов языков Фортран и БЕЙСИК. Следует отметить, что в большинстве ППП для ЕС ЭВМ программирование операций ввода и вывода данных предоставляется пользователю. Поэтому операторы ввода-вывода, а также некоторые другие вспомогательные операторы Фортрана системой не трапятся. Можно выделить следующие основные группы:

операторы Фортрана и БЕЙСИКА, практически совпадающие по синтаксису и семантике: присваивания, безусловного перехода GOTO, условного перехода IF, RETURN; операторы со сходной семантикой, но различающиеся по синтаксису: цикла, вычислительный GOTO, DIMENSION;

операторы Фортрана, не имеющие прямых аналогов в БЕЙСИКе, но достаточно просто моделирующиеся с помощью других операторов: арифметический IF, CONTINUE. К этой группе можно отнести также генерацию операторов NEXT для последнего оператора цикла; часто встречающиеся операторы Фортрана, для которых автоматический перевод на БЕЙСИК затруднен, но которые легко поддаются ручному переводу: PRINT, READ, WRITE, FORMAT, CALL, DATA;

редко используемые операторы Фортрана (особенно в программах для научно-технических расчетов): перехода по предписанию NAMELIST и присваивания значения метке BLOCK DATA и др.

Входные данные для системы ПАКТ — тексты программ на Фортране. Основные операторы Фортрана автоматически преобразуются в соответствующие им операторы БЕЙСИКА. Операторы, не поддающиеся автоматическому переводу, сохраняются в исходном виде (при распечатке текста выходной программы выделяются многоточием), хотя им присваиваются номера на БЕЙСИКе. Программа на Фортране, поступающая на вход инструментальной системы, не должна содержать синтаксических ошибок, так как это затрудняет ее перевод на БЕЙСИК. Основная часть синтаксических ошибок в переходах, условиях, циклах и некоторых других конструкциях языка выявляется системой ПАКТ. В таком случае некорректный оператор Фортрана печатается без преобразования.

Автоматизированный перевод программы на Фортране в программу на БЕЙСИКе с помощью системы ПАКТ осуществляется в три этапа. На первом этапе выполняются следующие преобразования: чтение исходной программы на Фортране; определение типа введенного оператора и частичная проверка правильности его записи; построение таблицы идентификаторов Фортрана, таблицы соответствия меток БЕЙСИКА и Фортрана и таблицы меток операторов цикла; частичное преобразование операторов Фортрана в операторы БЕЙСИКА с их записью в рабочий файл; генерация операторов NEXT при обнаружении меток, соответствующих таблице меток операторов цикла.

Для синтаксического анализа текста исходного модуля на Фортране используется подпрограмма, которая выделяет из оператора следующие элементы: идентификаторы, включая название оператора, вещественные числа, целые числа, в том числе метки, скобки, запятые, знаки операций, конец оператора.

Второй этап включает в себя: построение таблицы идентификаторов БЕЙСИКА с учетом типов переменных

(целых, вещественных, имен функций); печать таблицы идентификаторов и таблицы меток (для ручного перевода операторов READ, WRITE и т.п.); чтение операторов БЕЙСИКА из рабочего файла и их окончательную обработку (замена идентификаторов, меток и знаков операций); вывод подготовленной программы на БЕЙСИКе.

На третьем этапе пользователь вручную дорабатывает полученную программу на БЕЙСИКе: устанавливает ее связь с головной программой, задает необходимые размерности массивов, программирует операции ввода-вывода для решения конкретного класса задач.

Инструментальная система ПАКТ реализована на языке ПЛ/1 для операционной системы ОС ЕС (версия 6.1). С помощью системы ПАКТ выполнен перевод на язык БЕЙСИК известного пакета программ на Фортране (ЛНП-Ф), содержащего более 300 подпрограмм, которые охватывают широкий класс научно-технических задач [1, 2]. Уровень автоматизации при переводе одной подпрограммы указанного ППП на БЕЙСИК выполняется на ЭВМ ЕС1022 примерно за 1 мин.

Система ПАКТ разработана в двух модификациях, предназначенных для ПЭВМ «Искра 226», «Искра 1030», ЕС1840, ЕС1841. Возможна модификация системы и для других моделей ПЭВМ.

460352, Оренбург, пр. Победы, 13, ОНИЛ АСУ; тел. 7-98-56

ЛИТЕРАТУРА

1. Сборник научных программ на Фортране. Вып. 1. Статистика.— М.: Статистика, 1974.— 316 с.
2. Сборник научных программ на Фортране. Вып. 2. Матричная алгебра и линейная алгебра.— М.: Статистика, 1974.— 224 с.
3. Филлипс Д., Гарсия-Диас А. Методы анализа сетей: Пер. с англ.— М.: Мир, 1984.— 496 с.

Сообщение поступило 26.09.88

УДК 681.326:681.3.06

В. Г. Любимов, О. Н. Меламед, Е. С. Потапов, Д. В. Сеницын, В. М. Шилков

ОТЛАДОЧНЫЙ КОМПЛЕКС ДЛЯ ОДНОКРИСТАЛЛЬНЫХ ЭВМ «ЭЛЕКТРОНИКА С5»

Среди различных классов отладочных средств высшую производительность обеспечивают отладочные комплексы (ОК), позволяющие проводить совместную аппаратно-программную отладку микропроцессорных устройств (МПУ) на всех этапах проектирования [1]. Разработанный ОК предназначен для отладки и испытаний МПУ, построенных на 16-разрядных однокристалльных ЭВМ (ОЭВМ) «Электроника С5» [2]. Ряд особенностей ОК обусловлен спецификой системы команд семейства «Электроника С5» и структуры ОЭВМ.

ОК содержит основные типовые компоненты отладочных средств МПУ: базовый вычислительный комплекс (БВК), отладочный пульт, отладочное и фотосчитывающее устройство (ФСУ), блок питания. В качестве БВК выбран диалоговый вычислительный комплекс ДВК2М, возможен применение других комплексов с необходимым набором периферийного оборудования [3]. Отладочный пульт — функциональный модуль «Электроника С5-2107» — позволяет выполнять отладку МПУ в автономном варианте комплекса (без БВК). Отладочное устройство построено на основе специализированной ОЭВМ — отладочного кристалла, в котором дополнительно выведена буферизованная 16-разрядная внутренняя мультиплексированная шина адреса-данных. Применение специализированной ОЭВМ дает возможность отлаживать МПУ в полной системе команд «Электроника С5» и реальном масштабе времени.

Программа монитор обеспечивает следующие основные отладочные режимы:
 загрузку программы пользователя в ОЗУ ОК из БВК и ФСУ;

- передачу отлаженной программы в БВК;
- запись и чтение содержимого ОЗУ программ пользователя, регистров ОК, счетчика команд;
- выполнение программы пользователя по шагам;
- переход в режим точек останова;
- пуск программы с заданного адреса или после останова;
- выполнение программы в автоматическом режиме.

Кроме указанных режимов, ОК позволяет выполнять окончательную отладку МПУ на базе ОЭВМ, в ПЗУ которых занесены программы пользователя (пультовой режим комплекса). Для проверки состояния ОК в процессе эксплуатации используются программы контроля и дополнительный стенд, выполняющий прием-выдачу выходных входных сигналов комплекса.

Телефон 232-98-40, Ленинград

ЛИТЕРАТУРА

1. Весноватов М. Г., Домнин С. Б., Иванов Е. А., Муленко Л. Л., Широков Ю. Ф. Комплексные средства отладки микропроцессорных устройств // Электронная промышленность.— 1986.— Вып. 9 (157).— С. 3—8.
2. Кузнецов В. Я., Маслеников Ю. А., Никитин Э. А., Цветов В. П. Развитие микроЭВМ семейства «Электроника С5» и систем на их основе // Электронная промышленность.— 1979.— Вып. 11—12 (83—84).— С. 9—12.
3. Домнин С. Б., Иванов Е. А., Кушнир В. Д., Муленко Л. Л. Вычислительный отладочный комплекс «Электроника МС-0701» // Электронная промышленность.— 1986.— Вып. 2 (150).— С. 51—52.

Сообщение поступило 11.02.88

УДК 681.325.5—181.48

Б. Ч. Мусажалиев

ПУЛЬТ ОТЛАДКИ

Разработанный отладочный пульт (ОП) предназначен для систем, построенных на базе МПК БИС серии К580. Он заменяет штатный пульт оператора, имеет энерго-снабжение, общее с отлаживаемой системой (рис. 1), обеспечивает режимы прямого доступа к памяти (ПДП) для проверки аппаратных средств; отладки ПО (ОТЛ) для отладки прикладного программного обеспечения; работы в штатном режиме (РАБ).

В режиме ПДП пульт формирует все необходимые сигналы для управления любым абонентом, подключенным к системной магистрали. К особенностям режима можно отнести возможность управления как тактируемым, так и не тактируемым ЗУ, а также наличие звена дейзипели [1] для задания приоритета.

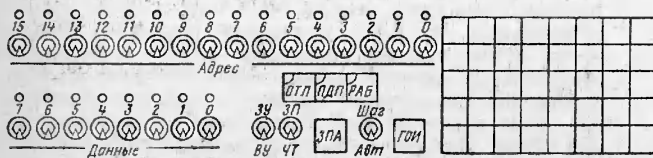


Рис. 1 Лицевая панель отладочного пульта

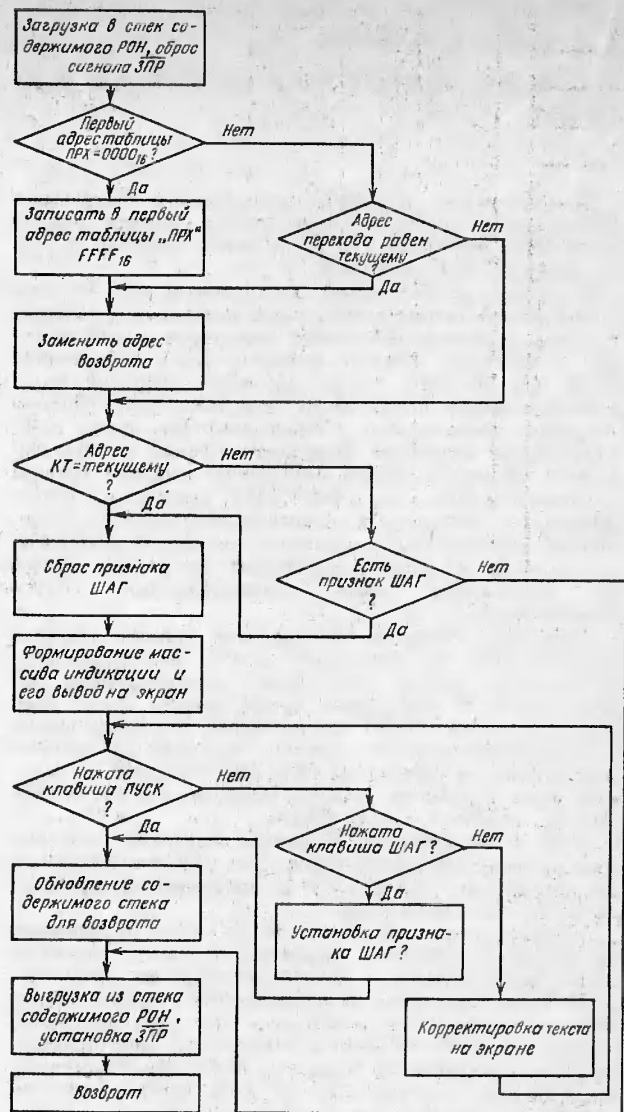


Рис. 2. Алгоритм подпрограммы обработки прерываний от монитора в режиме отладки

Режим ОТЛ поддерживает монитор объемом 2К, «прошитый» в ПЗУ. При подключении к проверяемой системе монитор размещается в свободной части адресного пространства и реализует подрежимы просмотра и модификации ЗУ — «ЗУ»; копировки файлов — «КПР»; программирования ПЗУ — «ПРГ»; установки контрольных точек — «КТ»; установки переходов, не предусмотренных прикладной программой, — «ПРХ». Режим ОТЛ в основном близок к описанному в работе [2].

Программы отлаживаются с использованием режима прерывания. Для этого один из выходов программируемого контроллера прерываний резервируется для подключения сигнала ЗПР, формируемого ОП. Прикладная программа в этом случае исполняется покомандно с проверкой соответствия текущего адреса программы одному из адресов в таблице контрольных точек или таблице переходов (рис. 2).

Сигнал ЗПР формируется под управлением монитора по схеме, приведенной на рис. 4. Подпрограмма обработки прерываний от монитора представлена на рис. 4.

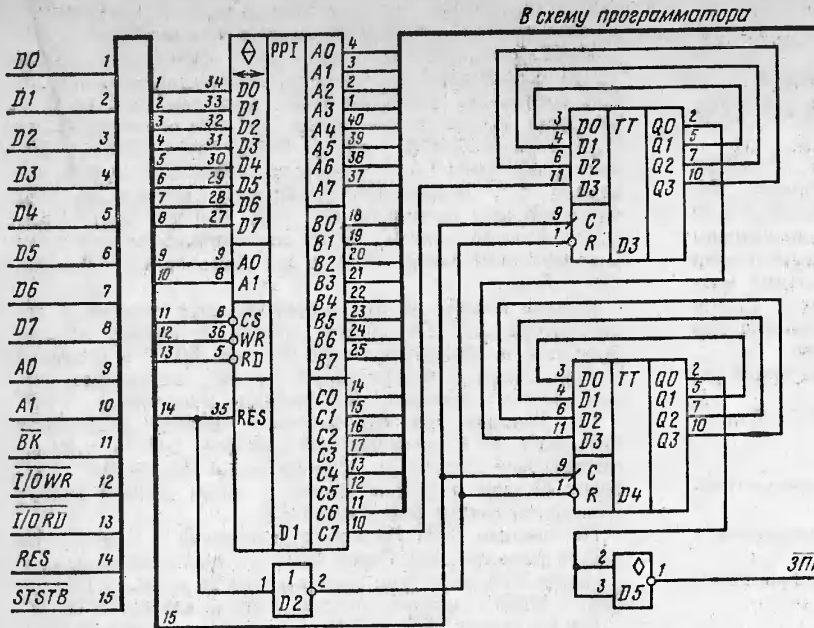


Рис. 3. Схема формирования сигнала ЗПР

В режиме «ПРГ» [3] программируются отдельные БИС ЗУ и блоки ПЗУ произвольного объема, для подключения которых в составе ОП предусмотрен специальный разъем.

Режим РАБ введен для удобства работы оператора; возможно покомандное (по машинным циклам) исполнение программы начиная с адреса, заданного тумблерами на лицевой панели ОП.

Телефон 25-96-24, Баку

ЛИТЕРАТУРА

1. Гибсон Г., Лю Ю.-Ч. Аппаратные и программные средства микроЭВМ.— М.: Финансы и статистика, 1983.— С. 130.
2. Дамке М. Операционные системы микроЭВМ.— М.: Финансы и статистика, 1985.— С. 24—49.
3. Дианов А. П., Шелкунов Н. Н. Модуль программирования микросхем ПЗУ // Микропроцессорные средства и системы.— 1985.— № 3.— С. 80—83.

Сообщение поступило 27.07.88

УДК 681.3

М. В. Черкашенко

АВТОМАТИЗИРОВАННАЯ ПОДГОТОВКА ПРОГРАММ ДЛЯ МИКРОКОНТРОЛЛЕРОВ

Разработан комплект программ для микроконтроллеров МБ 57.0*, позволяющий автоматизировать процесс их программирования, исходя из представления работы

* Разработаны Харьковским НПО САУ; имеют модификации по числу входов-выходов: ПК-36, ПК-64, ПК-128, ..., ПК-1024; применяются для управления приводами кузнечно-прессового, литейного оборудования, работотехническими комплексами и др.

** Оксененко А. Я., Диденко К. И., Черкашенко М. В. и др. Программирование микропроцессорных контроллеров для управления гидропневмоприводами.— М.: ВНИИТЭМР.— 49 с.

PUSH PSW ; Загрузка в стек слова состояния
MVI A, 001 ; Сброс сигнала ЗПР
OUT N ;
PUSH H ; Загрузка в стек содержимого PCH
PUSH D ;
PUSH B ;
 _____ ; Подпрограмма обслуживания для таблиц контрольных точек и переходов
POP B ; Восстановление содержимого
POP D ; PCH из стека
POP H ;
MVI A, 000 ; Установка сигнала ЗПР
OUT N ;
POP PSW ; Восстановления слова состояния
RET EI ; Разрешение прерываний
RET ; Возврат в обслуживающую программу

Рис. 4. Подпрограмма обработки прерываний от монитора

системы гидро- и пневмоприводов на формализованном языке.

В качестве последнего используется язык графов операций, описывающий параллельные и последовательные алгоритмы работы приводов. По графу операций составляются исходные данные «координатным способом» в матричной форме с учетом параллелизма или последовательности работы алгоритмов, вида исполнительных устройств приводов (с одно- или двухсторонним управлением), режимов работы системы приводов.

Для синтеза программ использован метод минимизирования числа инструкций**. Цель синтеза состоит в получении не общеприятного описания логической схемы системы приводов в виде системы логических уравнений в дизъюнктивной нормальной либо скобочной форме (она усложняет процесс технической диагностики системы, не наглядна и не удобна с точки зрения чтения программы непосредственно по графу операций), а минимизированной последовательности инструкций исходя из графа операций.

Тексты программ написаны на языке Паскаль и отлажены на ДВК2М в рамках ОС РАФОС, что дает возможность использовать комплект программ на мини-ЭВМ СМ3, СМ4, СМ1420, «Электроника 79» и др. без доработки. Исходная информация набивается на клавиатуре и корректируется на дисплее, затем записывается на магнитный диск. Далее дается команда на выполнение главной программы, которая вызывает в свою очередь ряд остальных процедур, записанных на магнитном диске и выполняющих синтез. Комплект занимает на диске объем около 50 Кбайт. Готовая программа с помощью программатора записывается в ОЗУ или ПЗУ микроконтроллера. Если необходима переналадка технологического оборудования, то следует поменять исходные данные, запустить комплект программ и получить соответствующую управляющую программу для микроконтроллера.

310141, Харьков, ул. Шатилова дача, 4, ВНИИГидропривод; тел. 45-83-53.

Сообщение поступило 21.09.88

ПРОГРАММАТОР ДЛЯ МИКРОСХЕМ СЕРИИ К573 НА БАЗЕ ПЭВМ «ИСКРА 226»

Программатор предназначен для работы с РПЗУ К573РФ4, К573РФ41, К573РФ5 и им подобными. Управляющая программа подготовлена на языке БЕЙСИК 02 ПЭВМ «Искра 226», позволяющем работать с оператором ввода-вывода и G10, дисковыми файлами, символьными массивами. В ОЗУ ПЭВМ организован символьный массив, работа с ячейками которого эквивалентна работе с РПЗУ, что дает возможность подготовить данные для записи в РПЗУ удобным и наглядным способом.

Программное обеспечение поддерживает следующие режимы:

- подготовки данных вручную с клавиатуры ПЭВМ и по программам пользователя;

- ввода-вывода данных на диск;

- копирования РПЗУ в область ПЭВМ, отведенную программой под «массив-копию».

- чтения «массива-копии» с представлением материала в удобной форме;

- программирования РПЗУ (перед программированием проверяется «чистота» РПЗУ);

- проверки качества записи;

- чтения РПЗУ с представлением результата в удобной форме.

Конструктивно программатор состоит из программируемого параллельного интерфейса (ППИ) и модуля персональности (МП). ППИ реализован в корпусе стандартного БИФ ПЭВМ «Искра 226» и соединен с МП кабелем.

Канал ПЭВМ имеет структуру типа «асинхронный бай-

товый интерфейс», что требует введения программно-доступных регистров адреса, данных и управления.

ППИ (рис. 1) включает узел дешифрации ФАУ (D17) с триггером адреса D4.1, обеспечивающий Лог 0 при совпадении значения ФАУ, выставляемого по С...5 разрядам ИМВВ с требуемым (можно реализовать на ПЗУ типа К556РТ4); буфер ИМВВ (D8, D9); буфер магистрали команд — МК (D1); дешифратор интерфейсных команд D7; формирователь сигнала «ответ» и кода интерфейсного состояния КВ (D5, V1, V2, D6.1, D6.2, D3.2); регистр данных D10 и три одинаковых регистра: управляющего слова (CW) и два 8-разрядных адресных (D11...D16).

Выдача команд по МК сопровождается стробом с дешифратора D7. По стробу данные с ИМВВ переписываются в соответствующий регистр. ППИ в ответ на любую команду, поступающую по МК, выставляет сигнал «ответ» и состояние КВ (команда выполнена).

По команде ВБ сбрасывается триггер управления передачей D4.2, переводящий регистр D10 в высокоимпедансное состояние. Одновременно строб ВБ переводит буферы в режим приема с шины данных РПЗУ; происходит чтение байта из РПЗУ.

По команде ПБП D4.2 устанавливается в 1, разрешая работу регистра D10. Строб ПБП, поступивший на вход С регистра D10, разрешает запись в регистр данных с ИМВВ; байт с ИМВВ устанавливается на шине данных ПЗУ.

По командам ППБ и ПБ в регистр младших и старших разрядов адреса записывается выводимый по ИМВВ адрес, а по команде ПК в регистр CW заносится управляющее слово. Импульс программирования и сигналы управления формируются программным путем через регистр CW. Источник программирующего напряжения управляется по сигналу MS (выв.8Тр.Упр. D8.2).

Разряды 0...2 CW подаются на входы \overline{WR} , \overline{CE} , \overline{OE} микросхемы К573РФ4 и \overline{CE} микросхемы К573РФ2 (рис. 2).

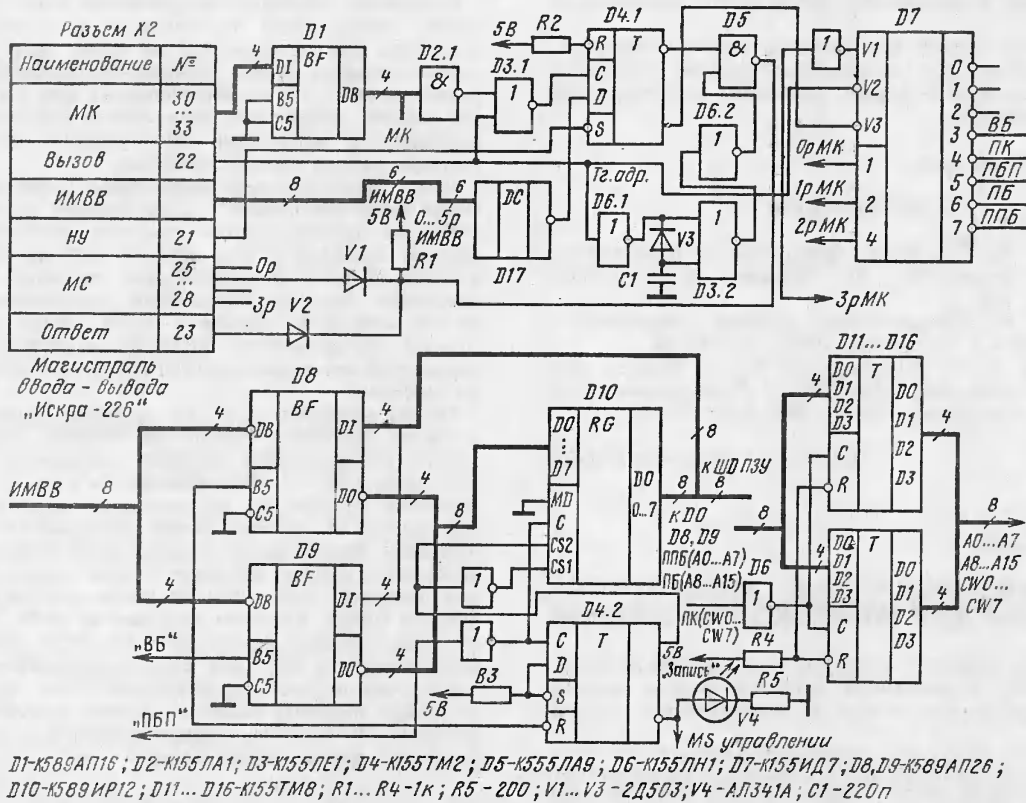


Рис. 1. Принципиальная схема ППИ

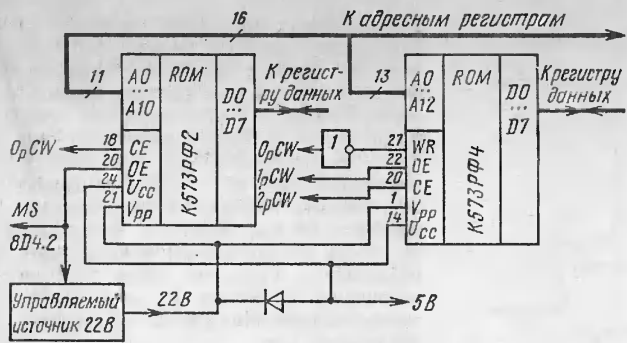


Рис. 2. Схема модуля персональноти

Процесс программирования осуществляется в следующем порядке:

- выводятся младшие разряды адреса 0...7;
 - выводятся старшие разряды адреса 7...15;
 - на шине данных РПЗУ выставляется байт данных;
 - в регистр CW последовательно записываются CW=04; CW=02; CW=03 (задержка 45 мс); CW=02; CW=04.
- При чтении РПЗУ вместо команды «ПБП» дается ВВ. CW всегда равно 00. В настоящее время разработана управляющая программа для работы с РПЗУ К573РФ2, К573РФ4, К573РФ41 и т. д. Наличие четырех программно-доступных регистров позволяет использовать ППИ, например с фотосчитывающим устройством типа FS-1501.

Телефон 592-97-28, Ленинград, Когель Александр Львович
Статья поступила 3.05.88

ОПЫТ РАЗРАБОТКИ АППАРАТНО-ПРОГРАММНЫХ СРЕДСТВ

УДК 681.325

Н. Н. Щелкунов, А. П. Дианов

СРЕДСТВА И МЕТОДЫ ПРОГРАММИРОВАНИЯ ОМК

(Продолжение цикла. Начало см. № 5 за 1987, № 1, 2 за 1988 и № 5 за 1989 гг.)

ОМК семейства MC48 [1] выполняются по обычной п-МОП или высококачественной Н-МОП технологии (табл. 1), что оказывает влияние на параметры сигналов программирования и чтения внутренней памяти программ (табл. 2). Функциональное назначение линий ОМК в режиме программирования приведено на рис. 1. Для адресации ультрафиолетом стираемого репрограммируемого ПЗУ (УФРПЗУ) объемом 1 или 2 Кбайт, встроенного в приборы семейства MC48, используют совмещенную шину адреса-данных AD0...AD7 — 8-разрядный порт BUS или D0...D7 (8741), а также младшую половину порта P2. Адрес запоминается в адресном регистре по фронту сигнала RESET (перепад напряжения с низкого значения на высокое). Шина AD0...AD7 служит не только для ввода младшей части 10-разрядного (или 11-разрядного для 8749) адреса A0...A9 (или A10), но и для ввода-вывода в ОМК программного кода. Свободные от ввода адреса входы младшей половины порта P2 должны либо иметь низкий уровень U_0

(уровень низкого потенциала ТТЛ-элементов), либо быть заземлены.

Приборы переводятся в состояние программирования и чтения подачи высокого потенциала $V_{EАН}$ (табл. 2) на вход EA (потребляемый ток $I \leq 1$ мА). Режим выбирается сигналом TO ($TO=U_0$ — программирование, $TO=U_1$ — чтение).

Питание логических схем ОМК в режиме программирования внутренней памяти осуществляется через вход V_{DD} (потребляемый ток $I \leq 20$ мА), на который подается напряжение V_{DDH} (табл. 2). Импульс программирования длительностью 50 мс (потребляемый ток $I \leq 1$ мА) поступает на вход PROG. Перепад его напряжения 0...+25 В в случае обычной и +5...+18 В в случае высококачественной технологии.

Для записи и чтения информации из внутренней памяти программ ОМК семейства MC48 с помощью системы программирования необходимо запустить встроенный в него генератор тактовых импульсов (ГТИ), что обеспечивает работу внутренней шины ОМК, передающей адреса и данные. Частота запуска OSC — 1...6 МГц при обычной технологии и 3...4 МГц — при высококачественной. Установка микросхе-

мы в панельку и запуск ГТИ осуществляется при условиях: $EA=+5$ В, $TO=U_0$, $RESET=U_0$, $V_{DD}=+5$ В и $PROG=0$ В (+5 В). Сигналы на входах портов BUS и P20...P23 не определены. Затем прибор переводится в состояние программирования и чтения подачи потенциала $V_{EАН}$ на EA.

Процесс программирования состоит в циклическом выполнении ряда операций, временные диаграммы которых приведены на рис. 2. На адресной шине устанавливается адрес, фиксируемый по фронту RESET, затем на освободившейся шине BUS — данные программирования D00...D07. После подачи питания V_{DD} на входе PROG генерируется импульс программирования (50 мс). Процесс завершается снятием напряжения V_{DD} и данных программирования D00...D07. Длительность фронта TR и среза TF сигналов на ответственных за программирование входах V_{DD} и PROG должна быть 0,5...100 мкс. После цикла программирования следует цикл контроля (чтение содержимого выбранной ячейки памяти программ и его сравнение с данными записи). Прибор переводится в состояние чтения ($TO=U_1$), а данные D10...D17 считываются с шины BUS. Цикл считывания завершается возвратом прибора в режим программирования ($TO=U_0$) и подачи на вход RESET

Таблица 1

Микроконтроллеры со встроенным УФРПЗУ

Тип		УФРПЗУ, Кбайт	Флажок секретности
прибора	технологии		
8748	п-МОП	1	Нет
K1816BE48	п-МОП	1	Нет
8741	п-МОП	1	Нет
8748H	Н-МОП	1	Нет
8749H	Н-МОП	2	Нет
8751H	Н-МОП	4	Есть
8744H	Н-МОП	4	Есть

Таблица 2

Параметры сигналов программирования и чтения MC48

Тип прибора	$V_{EАН}$, В (≤ 1 мА)	V_{DDH} , В (≤ 20 мА)	V_{PH} , В (≤ 1 мА)	OSC, МГц
8748	25	25	25	1...6
8741(A)	25(22)	25	25(22)	1...6
K1816BE48	25	25	25	1...6
8748H	18	21	18	3...4
8749H	18	21	18	3...4

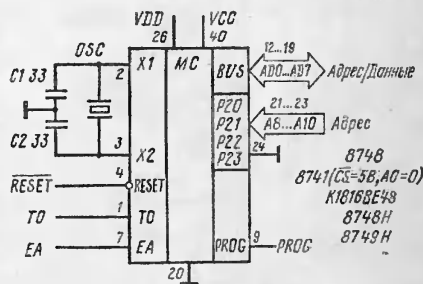


Рис. 1. Схема подключения ОМК MC48 в режимах программирования и чтения УФРПЗУ

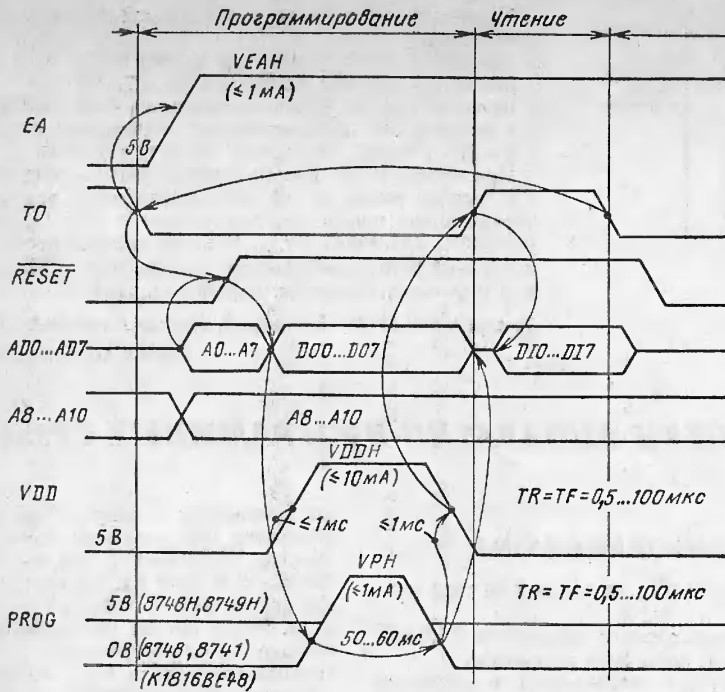


Рис. 2. Программирование и контроль УФРПЗУ MC48

дустановки и удержания сигналов на соответствующих входах прибора, которые на рис. 2 указаны стрелками без обозначения. Эти времена должны быть больше четырех периодов синхросигнала ALE программируемого прибора (60 периодов OSC). При частоте колебаний ГТИ в 1 МГц (минимально возможной) граничное значение этих времен — 60 мкс. Вместе с этим, длительность предустановки и удержания напряжения V_{DDH} на входе V_{DD} по отношению к фронту и срезу строба программирования PROG не должна превышать 1 мс.

В ОМК семейства MC48 предусмотрен специальный режим чтения внутренней памяти программ (рис. 3), с помощью которого можно контролировать состояние встроенной программной памяти, выполненной не только в виде УФРПЗУ, но и в виде масочного ПЗУ. К последнему классу ОМК относятся приборы 8048, 8048H, 8049H и 8050H, а также БИС K1816BE49.

Перевод в режим осуществляется подачей на вход EA напряжения +25 В для БИС и УФРПЗУ, выполненных по обычной технологии, и +12 В — по высококачественной, а также имеющих масочное ПЗУ.

В процессе чтения последовательно перебираются все адреса, поступающие в ОМК через порт BUS и младшую половину порта P2, и считываются данные D10...D17, выводимые из ОМК на шину BUS.

Младшая часть адреса A0...A7, вводимая через совмещенную шину адреса-данных BUS фиксируется по фронту сигнала RESET во внутреннем регистре прибора.

В случае УФРПЗУ на входе TO либо повторяют сигнал RESET (H-MOP технология), либо устанавливают постоянный уровень U₁ (n-MOP технология). При чтении масочной программируемого ПЗУ вход TO не используется. Времена предустановки и удержания всех сигналов те же, что и в режиме программирования УФРПЗУ.

В некоторых программаторах необходим автоматический контроль правильности установки прибора в панельку. Для этого применяется синхросигнал ALE (SYNC для 8041), появление которого воспринимается как признак правильной установки прибора.

Очистку памяти рекомендуется выполнять коротковолновым ультрафиолетовым светом с длиной волны 2000...3000 Å. Общая доза облучения должна быть не менее 15 Вт·с/см² при времени стирания около 15 мин. Приборы с УФРПЗУ следует устанавливать в непосредственной близости от источника света (2...3 см).

Следует заметить, что очистка памяти УФРПЗУ начинается при воздействии на него света с длиной волны менее 4000 Å. Эта длина волны входит в спектр солнечного света и ряда флуоресцентных ламп, поэтому кварцевое окошко нельзя оставлять открытым на длительное время. Прямой солнеч-

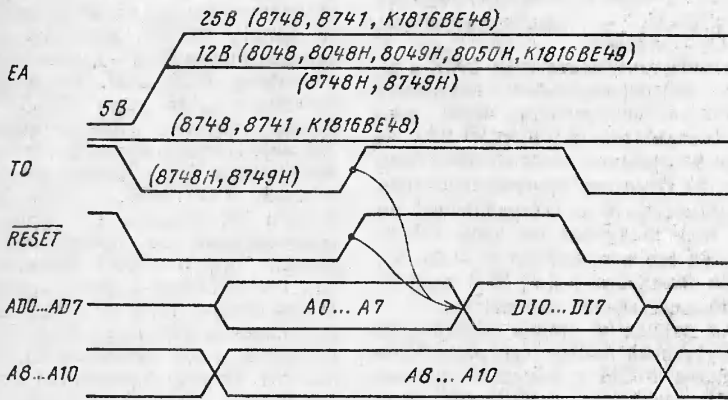


Рис. 3. Чтение УФРПЗУ MC48

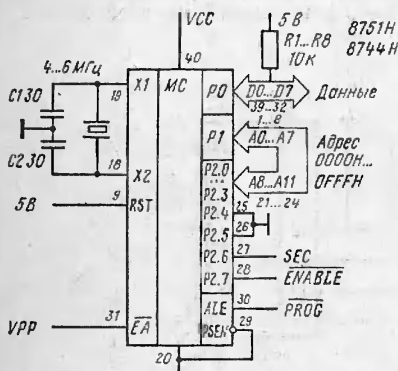


Рис. 4. Схема подключения ОМК MC51 в режимах программирования и чтения УФРПЗУ

низкого потенциала U₀. Далее циклы программирования и чтения повторяются.

В процессе программирования последовательно перебираются адреса внутренней памяти программ из диапазона 0000H...03FFH (07FFFH). К следующему адресу рекомендуется переходить только при устойчивом совпадении считанных данных D10...D17 и данных программирования D00...D07. Процесс завершается при выводе МК из режима программирования-чтения понижением потенциала на входе EA до уровня +5 В. Только в этом состоянии можно вынимать программируемый прибор из панельки программатора.

В процессе программирования-контроля следует выдерживать не только последовательность, но и времена пре-

ный свет способен вызвать очистку памяти ОМК всего за одну неделю, флуоресцентной лампе с уровнем освещенности рабочих помещений потребуется около трех лет.

В семейство MC51 (табл. 1) входят ОМК 8751Н и 8744Н, выполненные по Н-МОП технологии.

В отличие от MC48 в новых приборах для ввода адреса и обмена данными применяются отдельные шины адреса $A0...A11$ и данных $D0...D7$. Под ввод-вывод данных отводится 8-разрядный порт P0 (рис. 4). Особенность строения порта P0 — при чтении данных из внутренней памяти программ требуется внешняя нагрузка (рекомендуются резисторы 10 кОм, подключенные к источнику напряжением +5 В). При адресации памяти программ объемом в 4 Кбайт используются 8-разрядный порт P1 и младшая половина порта P2 (P2.0...P2.3). Две линии P2.4 и P2.5 резервируются для возможного расширения шины адреса (в существующих приборах их состояние не определено). На рис. 4 эти линии заземлены. Шины адреса и данных изолированы, поэтому не нужен дополнительный сигнал фиксации адреса (вход RESET в MC48).

В состоянии программирования и чтения внутренней памяти программ прибор переводится при $PSEN=0$ В и $RST=+2.5...+5.5$ В. Режим программирования или считывания определяется сигналом ENABLE, подаваемым на вход P2.7 и совместимым с ТТЛ-уровнями. При $ENABLE=U_1$ прибор находится в режиме программирования. При этом дополнительный сигнал SEC (вход P2.6), совместимый с ТТЛ-логикой, выбирает тип памяти программирования: память программ ($SEC=U_0$) или флажок секретности ($SEC=U_1$).

Для питания внутренних схем, ответственных за программирование, используется напряжение V_{pp} (+21 В, потребляемый ток ≤ 30 мА). Значение $V_{pp}=+21.5$ — предельное, его превышение приводит к порче прибора. Изменение напряжения должно происходить плавно, без скачков и выбросов. Импульс программирования PROG с низким уровнем активности и длительностью 50 мс подается на вход ALE. Параметры импульса PROG для приборов семейства MC51 совместимы с ТТЛ-логикой (в отличие от MC48).

До начала работы (по аналогии с MC48) должен быть запущен ГТИ на частоте 4...6 МГц при $PSEN=0$ В, $RST=+5$ В, $V_{pp}=+5$ В, $PROG=U_1$. Перед программированием УФРПЗУ необходимо убедиться, что $SEC=U_0$. Цикл программирования (рис. 5) начинается с установки потенциала U_1 на входе ENABLE (режим программирования); затем формируются адрес и данные, подается напряжение программирования V_{pp} и строб PROG (50 мс). Цикл завершается снижением напряжения программирования V_{pp} до уровня +5 В.

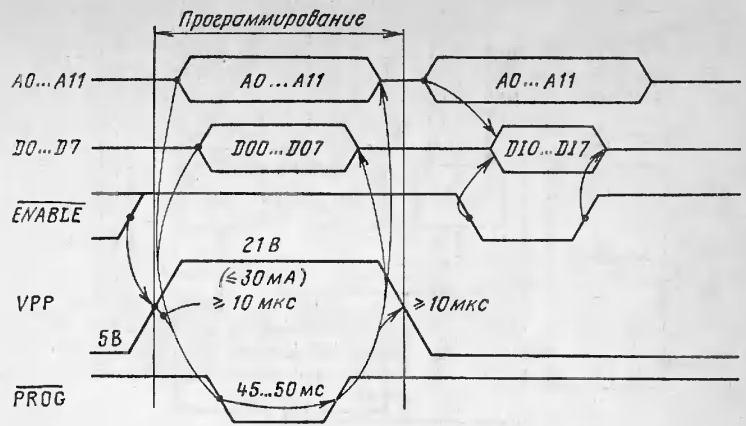


Рис. 5. Программирование и чтение УФРПЗУ MC51

Цикл чтения (рис. 5) начинается с установки адреса УФРПЗУ и генерации сто́рба чтения ENABLE.

Цикл может быть повторен любое число раз с любыми значениями адресов. После каждого цикла программирования (для его контроля) рекомендуется выполнять один цикл чтения. Переход к следующему адресу разрешается только при устойчивом совпадении данных программирования $D00...D07$ с действительными данными УФРПЗУ $D10...D17$. Для чтения памяти программ используют ряд циклов чтения с полным перебором всех адресов.

Времена предустановки и удержания (на рис. 5 указаны стрелками) должны быть не менее четырех периодов ALE или 48 тактов OSC (2 мкс при минимальной частоте работы ГТИ=4 МГц). Время предустановки и удержания высокого уровня напряжения V_{pp} по отношению к срезу и фронту импульса программирования PROG должно быть более 10 мкс.

В некоторых ОМК (табл. 1) имеется специальный флажок секретности — отдельная электрически программируемая одноквитная ячейка ПЗУ, стираемая ультрафиолетом. Его установка запрещает внешний доступ к внутренней памяти программ и не оказывает никакого воздействия на ее считывание схемами в обычном рабочем режиме.

Для программирования флажка секретности предусмотрена отдельная процедура, которая выполняется после записи и отладки программ пользователем во внутреннем УФРПЗУ. После этого доступ к УФРПЗУ в режиме программирования и чтения запрещен. Попытка сбросить флажок секретности засветкой ультрафиолетом приводит к автоматической очистке памяти программ. После засветки процедура программирования ОМК и его защита могут быть повторены необходимое число раз.

Процедура программирования флажка секретности 8751Н и 8744Н должна выполняться при $SEC=U_1$ (вы-

бор флажка секретности). Адрес и данные при этом во внимание не принимаются.

Технические средства программирования ОМК

Для программирования ОМК используются те же технические средства, что и для программирования ПЗУ, так как организация памяти программ ОМК аналогична организации автономных 8-разрядных ПЗУ. Гибкая микросистема МПС1 для записи информации в ПЗУ поддерживает широкий ассортимент программируемых ею приборов [2, 3]. Основа системы — одноплатный адаптер мМС8102 (рис. 6), формирующий специальный интерфейс проектирования ИИ8102, приспособленный для выполнения процедур программирования ПЗУ, ПЛМ и ОМК. В его состав входят до 40 линий параллельного ввода-вывода с ТТЛ-уровнями (8-разрядные шины $A1, B1, C1, B2$ и $C2$) и до пяти стробируемых выходов с независимо программируемыми уровнями напряжения ($E1...E5$), обеспечивающими в импульсном режиме ток нагрузки до 1 А. Для адаптера мМС8102 требуется набор простых терминальных модулей, учитывающих конкретную геометрию и особенности процедур программирования обслуживаемых ими приборов. Типовой набор для программирования ПЗУ (шесть модулей) предложен в [3]. Новый терминальный модуль мМС9414 (рис. 7) расширяет функции системы программирования МПС1, включая в состав обслуживаемых ею программируемых приборов ОМК типа 8751Н и 8744Н. В качестве двунаправленной шины данных используется 8-разрядный канал $A1$ адаптера мМС8102. Канал $B1$ и младшая часть канала $B2$ ($B20...B26$) отводятся под вывод адреса. Сюда же включены резервные адресные линии P2.4 и P2.5, а также линия выбора типа памяти SEC. С помощью младшей половины канала $C2$ создаются управляющие сигналы

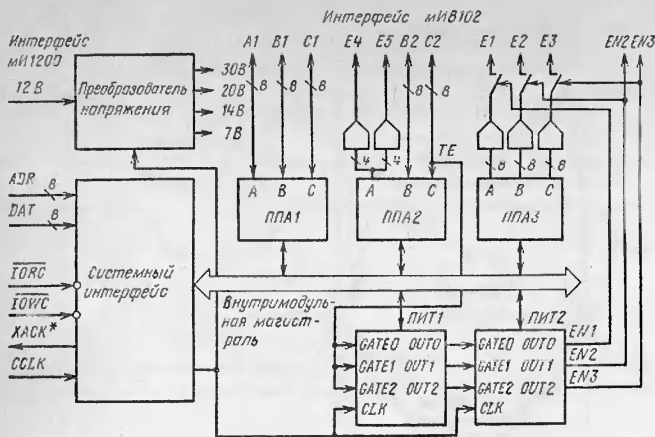


Рис. 6. Структурная схема адаптера мМС8102

RST, $\overline{\text{ENABLE}}$, $\overline{\text{PROG}}$ и PSEN. Источник E1 резервируется для формирования напряжения питания микросхемы V_{CC} (+5 В), а E2 — для напряжения программирования V_{PP}. Терминальный модуль программирования имеет встроенный кварцевый резонатор с частотой колебания 5 МГц и восемь нагрузочных резисторов R1...R8, подключенных к шине данных. Предусмотрен также светодиод VD1, зажигаемый источником E5. Он применяется для индикации рабочего состояния системы программирования.

Программное обеспечение программаторов ОМК

Адаптер мМС8102 позволяет для программирования ОМК использовать пакет MPPS [4], работающий в ОС ISIS-II, ДОС1800 и ДОС1810 (8-рядный режим работы). По аналогии с принципом построения аппаратуры программирования пакет программ MPPS состоит из двух частей: универсального ядра и набора терминальных файлов с процедурами обслуживания программируемых БИС. Их взаимодействие строго определено с помощью пяти простых процедур: BLANK,

WRITE, READ CHECK1 и CHECK2. Увеличение числа обслуживаемых пакетом БИС состоит в модификации старых и разработке новых терминальных файлов, для работы с модулем мМС9414 используется файл 87.ROM. Пакет MPPS обеспечивает: считывание с дисков загрузочных модулей для ОМК семейства MC51, подготовленных кроссовой системой программ КРОСС-51, и их загрузку в УФРПЗУ приборов 8751Н или 8744Н;

чтение содержимого внутренней памяти программ ОМК и его запись на диски системы проектирования МПСИ;

установку флажка секретности. Для поддержки функции программирования флажка секретности в составе обслуживаемых файлов 87.ROM приборов наряду с 8751Н и 8744Н введены два новых: 51Н.SEC и 44Н.SEC с программируемой памятью в 1 байт. Запись произвольной информации в эти приборы автоматически устанавливает флажок секретности ОМК 8751Н и 8744Н соответственно.

Разрабатываются технические и программные средства программирования микроконтроллеров MC48.

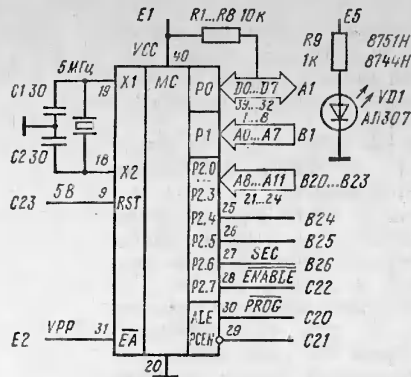


Рис. 7. Схема периферийного модуля мМС9414 для программирования ОМК 8751Н и 8744Н

Телефон 408-62-22, Москва
Техническая документация и ПО на средства программирования БИС поставляются центром НТТМ «Физтех»: 141700, Долгопрудный Московской обл., а/я 46.

ЛИТЕРАТУРА

1. Щелкунов Н. Н., Дианов А. П. Организация однокристалльных микроконтроллеров // Микропроцессорные средства и системы.— 1989.— № 5.
2. Дианов А. П., Щелкунов Н. Н. Система проектирования микропроцессорных устройств // Микропроцессорные средства и системы.— 1987.— № 5.— С. 83—86.
3. Дианов А. П., Щелкунов Н. Н. Модули программирования логических схем // Микропроцессорные средства и системы.— 1988.— № 1.— С. 40—44.
4. Щелкунов Н. Н., Дианов А. П. Организация программного обеспечения программаторов ПЗУ // Микропроцессорные средства и системы. 1988.— № 2.— С. 29—32.

Статья поступила 11.01.87

УДК 681.325

И. В. Музалевский, А. В. Овчинников, Э. П. Розенштейн

ПОСТРОЕНИЕ АРБИТРОВ НА ПЗУ

В любой вычислительной машине, имеющей несколько процессоров, использующих общую память, возникает задача организации режима доступа к ней. Каждый процессор выставляет запрос на доступ; запрос обрабатывается устройством управления запросами — арбитром и доступ либо предоставляется, либо нет [1, 2]. В большинстве случаев алгоритм обслуживания запросов достаточно прост и является либо приоритетным (по наибольшему приоритету), либо беспriorитетным (по очереди).

Простой, универсальный и быстродействующий арбитр можно построить на программируемых логических устройствах, например на ПЗУ [3—5]. С помощью ПЗУ легко

реализуются устройства управления, генерирующие определенные управляющие сигналы, счетчики с различными и переменными коэффициентами пересчета и т. п. Недостаток ПЗУ — существенное потребление мощности. При построении автоматов необходимо помнить и о таком недостатке, как «звон», представляющий собой ложное срабатывание микросхемы на выходах при изменении кода на адресных входах во время активного сигнала «Выбор кристалла».

Предположим, что четыре устройства с приоритетами 0, 1, 2, 3 требуют доступа к памяти; приоритет 0 — наибольший, 3 — наименьший. Тогда соотношение кодов на адресных входах ПЗУ и содержимое ячеек будут соответствовать табл. 1.

Если устройства работают синхронно, то для устранения звона на выходе ПЗУ ставится специальный буфер (рис. 1). В асинхронном режиме работы возможна

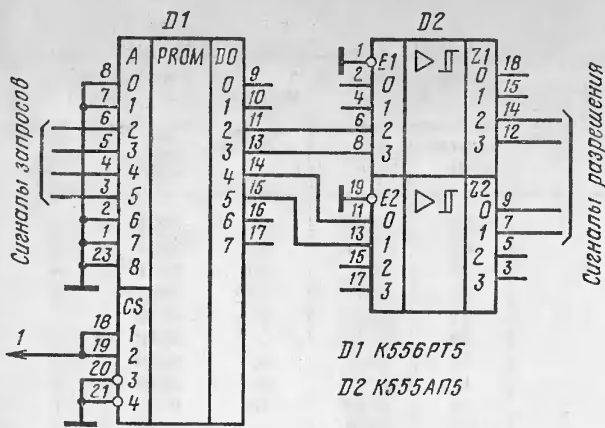


Рис. 1. Схема подавления звона на выходе ПЗУ

Обязательные запросы от устройства*	Адрес ячейки ПЗУ	Двоичный код запроса на выходах ПЗУ	Содержимое ячейки
1	XX0001**	00	0001
2	XX0010	01	0010
3	XX0100	10	0100
4	XX1000	11	1000
Нет запроса	XX0000	00	0000

* Наличие запросов от других устройств необязательно.
 ** На месте X — 0 или 1 (четыре комбинации).

Таблица 1

Запросы устройств*	Адрес ячейки ПЗУ	Содержимое ячейки
Запросов нет	0000	0000
1	0001	0001
2	0010	0010
3	0100	0100
4	1000	1000
1, 2	0011	0001**
1, 3	0101	0001**
1, 4	1001	0001**
1, 2, 3	0111	0001**
1, 2, 4	1011	0001**
1, 3, 4	1101	0001**
2, 3	0110	0010***
2, 4	1010	0010***
2, 3, 4	1110	0010***
3, 4	1100	0100****
1, 2, 3, 4	1111	0001**

* Запрос устанавливается высоким уровнем и держится все время обмена с памятью. ** Наибольший приоритет устройства 1; *** — устройства 2; **** — устройства 3.

ситуация, когда во время обмена с памятью одного из устройств приходит запрос от устройства с более высоким приоритетом. Тогда в соответствии с табл. 1 сигнал разрешения выдается устройству с более высоким приоритетом, а для устройства с меньшим приоритетом снимается, несмотря на то, что обмен с памятью не закончен. Таким образом, возникает необходимость фиксации текущего сигнала разрешения и блокирования всех запросов, кроме текущего, на время обмена. Решить этот вопрос можно без введения дополнительной аппаратуры, используя обратную связь (рис. 2), представленную двоичными кодами текущего запроса: 00, 01, 10, 11, которые устанавливаются на адресных входах ПЗУ и выходе буфера одновременно с сигналами разрешения. В этом случае таблица истинности ПЗУ будет выглядеть, как табл. 2.

Режим работы арбитра

В исходном состоянии запросы отсутствуют, адрес ячейки ПЗУ равен 00000₂, содержимое этой ячейки и, соответственно, комбинация сигналов разрешения — 0000₂ плюс двоичный код запроса 00₂.

Предположим, пришел запрос от устройства 2. Тогда в первый момент времени адрес ячейки ПЗУ равен 000010₂, адрес ячейки ПЗУ станет равным 010010₂ (см. табл. 2). Сочетание сигналов разрешения на выходах ПЗУ при этом не изменится. Возникающий звон сглаживается буфером. В случае прихода любого из трех запросов (от устройств 1, 3, 4) или их комбинации сочетание битов на выходах ПЗУ и, соответственно, двоичный код запроса и сигналы разрешения также не изменятся.

После снятия запроса устройством 2 возможны три режима дальнейшей работы арбитра. **Режим 1.** Если к моменту снятия запроса других запросов не пришло, то адрес ячейки ПЗУ станет равным 010000₂. Через время τ на выходах ПЗУ установится комбинация 000000₂. **Режим 2.** Если пришел запрос, например от устройства 3, то адрес ячейки ПЗУ равен 010100₂. Через время τ на выходах ПЗУ установится комбинация 100100₂. **Режим 3.** Если пришли запросы от нескольких устройств, например от устройств 3 и 4, то адрес ячейки ПЗУ равен 011100₂. Через время τ на выходах ПЗУ установится комбинация 100100₂ так как устройство 3 имеет более высокий приоритет, чем устройство 4. В случаях прихода запроса от устройств 1, 3 или 4 арбитр работает аналогично.

Рассмотрим работу арбитра в интервале времени t . Предположим, что в начальный момент времени, когда на адресных входах ПЗУ установлены биты 000000₂, пришел запрос от устройства 2. Ответная реакция на выходах ПЗУ (код 010010₂) появится через время τ_1 , а сигналы разрешения на выходе буфера и двоичный код на адресных входах — спустя время τ_2 ($\tau = \tau_1 + \tau_2$). Предположим, что по истечении времени τ_3 ($\tau > \tau_3 > t$ звона), прошедшего с момента прихода запроса от устройства 2, приходит

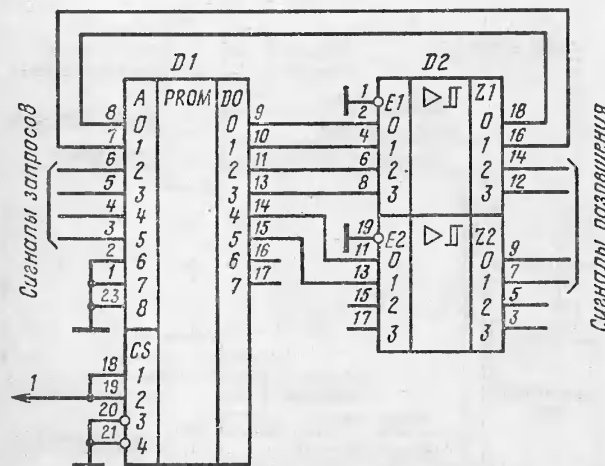


Рис. 2. Схема блокирования всех запросов на время обмена с памятью

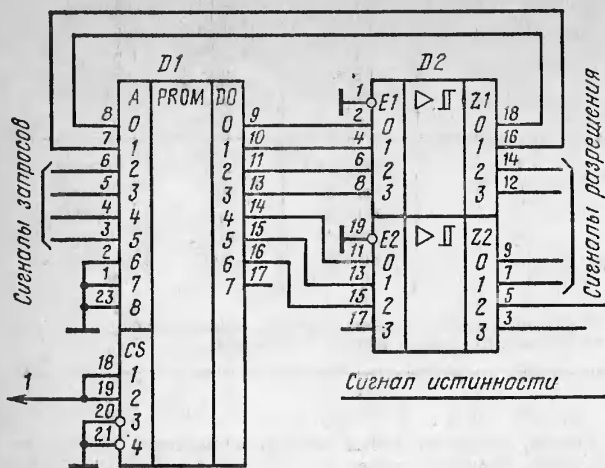


Рис. 3. Схема защиты от ложного срабатывания

Таблица 3

Обязательные запросы от устройства	Адрес ячейки ПЗУ		Двоичный код запроса на выходе	Содержимое ячейки	
	Двоичный код запроса на входе	Запрос		Код разрешения	Бит истинности
1	00	XXX1*	00	0001	1
	01	XX01	00	0001	0
	10	X0X1	00	0001	0
	11	0XX1	00	0001	0
2	01	XX1X	01	0010	1
	00	XX10	01	0010	0
	10	X010	01	0010	0
3	11	0X1X	01	0010	0
	10	X1XX	10	0100	1
	00	X100	10	0100	0
4	01	X100	10	0100	0
	11	0100	10	0100	0
	11	1XXX	11	1000	1
Нет запроса	00	1000	11	1000	0
	01	1000	11	1000	0
	10	1000	11	1000	0
	11	1000	11	1000	0
Нет запроса	XX	0000	00	0000	0

Обязательные запросы от устройства	Адрес ячейки ПЗУ		Двоичный код запроса на выходе	Содержимое ячейки	
	Двоичный код	Запрос		Код разрешения	Бит истинности
1	00	XXX1*	00	0001	1
	01	XX01	00	0001	0
	10	X001	00	0001	0
	11	0X01	00	0001	0
2	01	XX1X	01	0010	1
	00	XX10	01	0010	0
	10	X01X	01	0010	0
3	11	0X1X	01	0010	0
	10	X1XX	10	0100	1
	00	X100	10	0100	0
4	01	X100	10	0100	0
	11	0100	10	0100	0
	11	1XXX	11	1000	1
	00	1000	11	1000	0
	01	1000	11	1000	0
	10	1000	11	1000	0

* На месте X — 0 или 1.

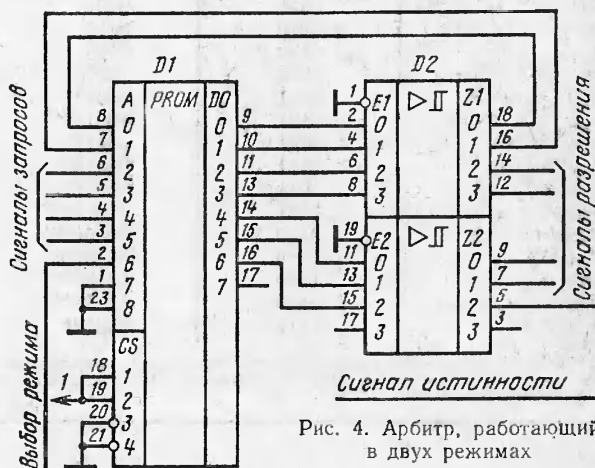


Рис. 4. Арбитр, работающий в двух режимах

* На месте X — 0 или 1.

запрос от устройства 1, имеющего более высокий приоритет. Тогда арбитр должен оставить обработку запроса от устройства 2 и заняться запросом от устройства 1. Произойдет ложное срабатывание арбитра, появится ложный сигнал разрешения для устройства 2. Длительность ложного срабатывания будет равна $\sim t_3$. Если ПЗУ имеет задержку $\tau_1=100$ нс, то t_3 может достигать также 100 нс и более (за счет t_2).

Избавиться от ложного срабатывания можно, введя в арбитр выходной сигнал истинности, свидетельствующий об истинности комбинации сигналов разрешения и инициирующий память к началу обмена с устройством (рис. 3). Этот сигнал возникает только тогда, когда двоичный код запроса установлен на адресных входах ПЗУ одновременно со «своим» запросом (табл. 3). Работа арбитра рассматривается в предположении, что коды системы разрешения представлены в позиционном виде. В общем случае они могут быть любыми. Например, представление кода разрешения в двоичной форме (табл. 3) удобно для управления мультиплексорами К555КП12, дешифраторами К555ИД4, другими устройствами.

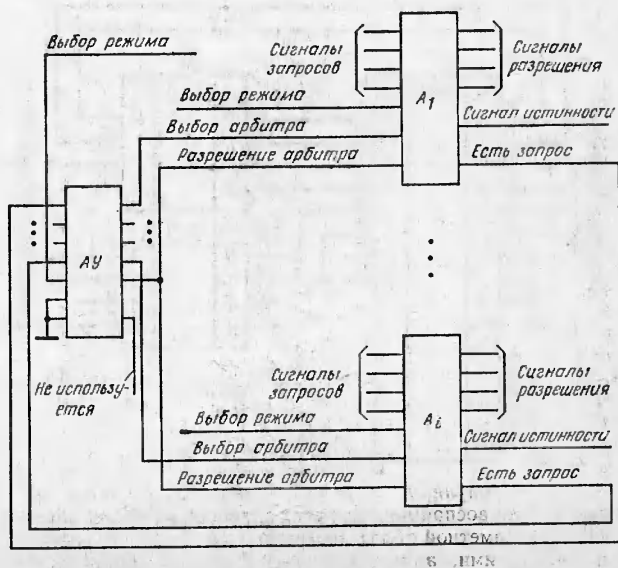


Рис. 5. Схема совместного включения нескольких арбитров

Изменить режим работы арбитра можно, предусмотрев специальный вход выбора режима. Допустим, что арбитр работает по описанному выше алгоритму в течение времени T_1 , а T_2 — по алгоритму, учитывающему следующее распределение приоритетов устройств 1...4: 1, 0, 2, 3. В этом случае он работает в соответствии с табл. 3, если на входе выбора режима установлен 0, и в соответствии с табл. 4, если — 1 (рис. 4). Получить арбитр, способный к наращиванию, можно, введя входы «Выбор арбитра» и «Разрешение арбитра» и выход «Есть запрос» (рис. 5). Арбитр АУ управляет работой арбитров A_i . Он выставляет сигнал «Есть запрос», который, в свою очередь, является запросом для АУ. Последний в соответствии с приоритетом устанавливает сигнал разрешения, который для одного из A_i служит сигналом «Выбор арбитра», затем сигнал истинности, который для всех A_i является сигналом «Разрешение арбитра». После получения этих двух активных сигналов A_i начинает работать так, как было описано выше.

Минимальное время обработки запроса арбитром по схеме, представленной на рис. 4, равно 2τ ; по схеме на рис. 5 — 5τ . Если задержка ПЗУ $\tau_1=100$ нс и буфера $\tau_2=20$ нс, то это время соответственно равно 240 и 600 нс.

Телефон 133-80-89, Москва, Овчинников А. А.

ЛИТЕРАТУРА

1. Тербер К. Дж. Архитектура высокопроизводительных вычислительных систем. — М.: Наука, 1985.
2. Снижение соперничества при доступе к совместно используемой памяти в многопроцессорных системах // Экспресс-информация. Сер. Вычислительная техника. — 1988. — № 4.
3. Баранов С. И., Скляров В. А. Цифровые устройства на программируемых БИС с матричной структурой. — М.: Радио и связь, 1986.
4. Вайда Ф., Чакань А. МикроЭВМ. — М.: Энергия, 1980.
5. Стрижов Р. В., Пашкевич С. В. Разрешение конфликтной ситуации при работе двух процессоров с общим полем оперативной памяти // Микропроцессорные средства и системы. — 1988. — № 2.

Статья поступила 4.10.88

УДК 681.3.06

А. Ю. Кучко, Д. В. Лопухов

ПРОГРАММНЫЕ СРЕДСТВА ВВОДА-ВЫВОДА РЕЧЕВЫХ СИГНАЛОВ ДЛЯ МИНИ-ЭВМ СМ1420

Программные модули, предназначенные для ввода-вывода в реальном масштабе времени речевых сигналов, написаны на языке MACRO 11, работают под управлением однозаданного SJ-монитора ОС РАФОС и позволяют выполнять следующие функции:

ввод речевого сигнала с кодера в расширенную память ЭВМ;

вывод речевого сигнала из расширенного ОЗУ на декодер;

многократный вывод речевого сигнала на декодер с заданной паузой между циклами вывода (вывод в кольцо); запись заданного объема данных в расширенном ОЗУ на магнитный диск;

считывание данных с магнитного диска в заданную область расширенного ОЗУ;

последовательное отображение содержимого расширенного ОЗУ на экране осциллографа (визуализация).

Объем памяти, доступный для ввода-вывода, равен 192 Кбайт. Частота дискретизации аналогового сигнала не превышает 20 кГц.

Выбор режима обмена и настройка его на конкретную функцию (вывод всего объема данных из ОЗУ на декодер, вывод сегмента речи и т. д.) осуществляются оператором в диалоговом режиме «вопрос-ответ». Выход из режима происходит при нажатии произвольной клавиши на консоли. Выполнение всех функций сопровождается контролем ошибок. При возникновении ошибочных ситуаций выдается соответствующее сообщение и работа модуля прекращается.

Любой программный модуль может быть ориентирован как на послонный, так и на побайтовый режимы обмена между интерфейсом кодера и расширенным ОЗУ. Система открытая, поэтому допускает включение новых программных модулей в основную программу.

103064, Москва, НИИР; тел. 261-14-17.

Сообщение поступило 17.02.88

ЭКСПЕРТНЫЕ СИСТЕМЫ

решении практических задач на уровне высококвалифицированного эксперта [4, 5].

ЭС должны обладать следующими характерными особенностями:

получать заранее не известный алгоритм (строится самой ЭС с помощью рассуждений, базирующихся на логических выводах и эвристиках);

анализировать и объяснять свои действия и выводы в терминах, понятных пользователю;

располагать средствами диалога с пользователем или экспертом, не являющимся специалистом;

приобретать новые знания с изменением поведения.

В ЭС принято выделять четыре существенные компоненты: БЗ, машину логического вывода, модуль извлечения знаний и систему объяснения.

Первые ЭС появились в начале 70-х годов в тех отраслях деятельности человека, где знания неформализованы (медицина) или частично формализованы (химия, геология и т. д. [6]). Достаточно подробный перечень современных ЭС и соответствующая библиография приведены в [7—10].

Широкий интерес различных категорий пользователей к ЭС, постоянное увеличение производительности и объ-

УДК 681.3.022

С. Х. Айтъян, К. Р. Гуарян

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ЭКСПЕРТНЫХ СИСТЕМ НА ПЭВМ

Экспертные системы (ЭС) разрабатываются для широкого спектра проблем диагностики, проектирования, планирования, управления, интерпретации, прогнозирования, там где применение вычислительной техники ранее считалось неэффективным [1]. Такие задачи отличаются сложностью описания в цифровой форме или задания формальных целей. Поэтому в ЭВМ закладываются не формальные алгоритмы, а прагматические правила экспертов, составляющие знания системы, по которым она принимает решения в сложных ситуациях. Знания организуются в базу знаний (БЗ), имеющую определенную структуру, механизмы доступа и алгоритмы использования [1—3].

Растущая популярность ЭС во многом объясняется их способностью воспринимать знания специалистов в определенной предметной области, обеспечивать доступ и манипулирование ими, а также выдавать рекомендации при

мов оперативной и внешней памяти ПЭВМ стимулировали стремительный рост числа ЭС, ориентированных на ПЭВМ. Большинство ЭС разрабатывается для ПЭВМ, совместимых с IBM PC XT/AT или PS/2 [11, 12].

Для проектирования ЭС используются универсальные и проблемно-ориентированные инструментальные средства (ИС). К основным характеристикам ИС относятся: особенности представления знаний о предметной области в БЗ ЭС и способ вывода заключений в ходе консультации (способ рассуждений); средства работы в условиях неопределенности; интерфейс пользователя, наличие и особенности компонентов объяснения; интерфейс разработчика ЭС, средства редактирования БЗ и отладки — окружение разработки; системный интерфейс — средства связи с внешним по отношению к ИС и ЭС системным и прикладным программным обеспечением.

Средства проектирования ЭС

В настоящее время сложилась своего рода иерархия ИС, различающихся между собой числом заранее скомпонованных элементов ЭС и требуемой квалификацией разработчика:

1. Традиционные алгоритмические языки программирования и специальные программные среды.
2. Функциональные, объектно-ориентированные языки и языки логического программирования.
3. Языки представления знаний.
4. ИС для разработки ЭС системного уровня.

Среди наиболее часто применяемых в качестве ИС традиционных алгоритмических языков выделяют Паскаль, С, БЕЙСИК, Фортран. Специальные программные среды объединяют быстрые однопроходные компиляторы, экранные редакторы и интерактивные средства отладки: Turbo-Pascal, Turbo-C. Эта категория ИС позволяет создавать наиболее действующие, но и наиболее трудоемкие ЭС для ПЭВМ.

Функциональные (Лисп), объективно-ориентированные (Смолток) и языки логического программирования (Пролог) дают возможность достаточно просто моделировать БЗ и машину логического вывода ЭС. Пролог основан на языке формальной логики предикатов и обеспечивает представление знаний в виде фактов и правил вывода. В него встроена конкретная стратегия вывода — обратный вывод в Хорновском исчислении предикатов.

Реализация языков Лисп и Пролог на существующих моделях ПЭВМ не дает высокой производительности традиционных языков программирования, однако возможность быстрого создания прототипов ЭС на них все чаще привлекает внимание разработчиков. Этому в немалой степени способствует разработка быстрых компиляторов с языка Лисп — мю-Лисп и, в особенности, программная среда фирмы Borland — TurboProlog. В настоящее время идет оживленная дискуссия о применимости различных языков программирования для разработки ЭС [5, 13]. Думается, что наиболее ярко достоинства языков Лисп и Пролог проявятся при широком внедрении специализированных Лисп- и Пролог-машин.

Использование рассмотренных языков программирования в качестве ИС оправдано на заключительной стадии разработки ЭС — создании коммерческого продукта. В остальных случаях более предпочтительны **специальные инструментальные средства (expert systems development tools)**, позволяющие проектировать ЭС на более высоком уровне абстракции с учетом понятия конкретной предметной области.

Другой класс ИС — **языки представления знаний** — содержат механизмы представления знаний и управления. К ним можно отнести OPS5+ (Computer Thought Corporation), Topsy (Dynamic Master Systems), Expert OPS5 (Expertelligence), OPS83 (Production Systems)

[14]. Основной язык среди них — OPS5 — обеспечивает потребности инженерии знаний, но не отвечает конкретным стратегиям решения задач или схемам представления знаний.

ИС для разработки ЭС системного уровня — «оболочки» (shells) и инструментальные среды проектирования (environments) — включают основные компоненты ЭС и позволяют учитывать специфику конкретной области экспертизы, не тратя усилий и времени на общесистемные задачи.

Оболочки дают возможность работать на системном уровне, т. е. уровне понятий предметной области; при этом основная роль разработчика заключается в наполнении БЗ знаниями экспертов. Оболочки содержат невидимые для пользователя структуры данных и управляющую стратегию, необходимую для каждого конкретного приложения. Благодаря этому объем программирования становится минимальным, но за это приходится платить потерей гибкости в случае, если приложение не относится к классу задач, которые имелись в виду при создании конкретной оболочки [5]. Оболочки появились как развитие «пустых» (empty) ЭС. Первые пустые ЭС — meta Dendral и EMycin появились в проектах Dendral и Mycin и имели фиксированные модели представления знаний и рассуждений над знаниями. Это наиболее быстрый способ создания ЭС в том случае, если разработчика полностью устраивают проектные решения, принятые в конкретной пустой ЭС.

Обычно оболочки строились на основе существующих ЭС, например Mycin или Prospector [1, 7, 15]. При этом решения задач ограничиваются конструктивными подходами, заложенными в исходных ЭС. С точки зрения пользователей ПЭВМ, эта категория ИС представляет в настоящее время наибольший интерес в связи с массовым внедрением методов и средств инженерии знаний во многие отрасли человеческой деятельности. Именно к ней относится большинство коммерческих ИС создания ЭС на ПЭВМ.

ИС для разработки ЭС системного уровня подразделяются на средние (mid-size tools) и малые (small tools). Средние ИС, представляющие собой технологическую среду поддержки проектирования ЭС, обычно предполагают наличие широкого спектра используемых инструментальных программ, ориентированных на высококвалифицированных специалистов и инженеров по знаниям. С помощью таких ИС могут быть созданы как ЭС в целом, так и отдельные функциональные компоненты ЭС: БЗ, машина вывода, пользовательский интерфейс и т. д. Такие системы работают на ПЭВМ типа IBM PC AT, PS/2 или рабочих станциях SUN, APOLLO.

Малые ИС, являющиеся по своей сути оболочками ЭС, предназначены для пользователей-непрофессионалов в области искусственного интеллекта и ЭС. Разработчику ЭС достаточно наполнить БЗ оболочки знаниями, относящимися к его предметной области, и описать интерфейс пользователя с ЭС. Конечно, в этом случае пользователь вынужден применять формализмы представления знаний и особенности функционирования машины вывода имеющейся оболочки. В этой связи выбор ИС, подходящего для задач конкретного пользователя, приобретает решающее значение.

Провести четкую грань между средними и малыми ИС довольно сложно, поскольку многие фирмы, разрабатывающие ИС для ЭС, оснащают оболочки альтернативными возможностями по представлению знаний (правила, фреймы, метаправила), работе с нечеткими знаниями, использованию различных процедур вывода (прямой и (или) обратный вывод).

Сведения о некоторых ИС для ПЭВМ, совместимых с IBM PC и функционирующих под управлением ОС MS DOS [11, 12, 16—21], приведены в табл. 1.

Малые коммерческие ИС предназначены для разработки малых и средних ЭС (200...300 правил) пользователями, которые чаще всего сами занимаются эксплуатацией ЭС. Наиболее популярны в этом классе ИС VP Expert, Personal Consultant Easy, Level Five (Insight) [12], Crystal [22], Expert Edge, Xi Plus.

Коммерческие ИС системного уровня

Название (фирма)	Модель представления знаний	Метод вывода	ПЭВМ, язык реализации
Advisor-2 (Expert Systems International)	Правила	—	PC XT/AT
APES (Programming logic Systems)	Логика	—	PC XT, C
Copernicus (S.I) (Icknowledge/Framentec)	Правила, фреймы	—	Pascal PC AT, C
Crystal (Intelligent Environments)	Правила; байесовская процедура для нечетких знаний	Обратный вывод	PC XT, C
ESP Advisor (Expert Systems International)	Правила	Прямой и обратный выводы	PC XT, Prolog-2
Expert Edge (Helix Expert Systems)	Правила, иерархия объектов	Обратный вывод	PC, XT, C
ESP Frame Engine (Expert Systems International)	Фреймы, правила, демоны, наследование свойств в иерархии фреймов	Прямой и обратный выводы	PC AT, Prolog-2
Experkit II (Act Informatique)	Правила, пакеты правил	Прямой вывод	PC AT, LeLisp
Exsys/Exsys Professional (Exsys)	Правила	Обратный вывод	PC XT, C
Extran 7.2 (Intelligent Terminals)	Множество примеров, деревья принятия решений	Прямые и обратные выводы, вывод по индукции	PC XT/AT, Fortran 77
GoldWorks (GoldHill Computers)	Фреймы, правила, объекты	—	PC AT, Lisp
GURU (MicroData Basa Systems)	Правила, факторы уверенности для работы с нечеткими знаниями	Прямой и обратный выводы	PC XT/AT, C
Humble (Xerox Specia Info. Systems)	Ориентировано на объекты	—	PC XT, Smalltalk
KDS 2&3 (KDS Corporation)	Множество примеров, правила	Прямой и обратный выводы, индукция	PC XT, assembler
Kiss (Brainware)	Множество примеров	Обратный вывод и индукция	PC XT, MicroProlog
KnowledgePro (Knowledge Garden)	Правила	—	PC XT, TurboPascal
Leonardo (Creative Logic)	Правила, фреймы, процедуры; байесовские процедуры, факторы уверенности, нечеткая логика	Прямой и обратный выводы	PC XT/AT, Fortran, assembler
Level 5 (Insight) (Information Builders)	Правила	—	PC XT, Pascal
M.1 (Teknowledge/Framentec)	—»—	Обратный вывод	PC XT, Prolog, C
MicroExpert (McGraw-Hill Book Company)	—»—	—»—	PC XT, Pascal
Nexpert (Newron Data)	—»—	—	PC XT, assembler
Nexpert Object (Newron Data)	Правила, гипотезы	Прямой и обратный выводы	PC XT, C
Nexus (MindSoft)	Правила, процедуры	—»—	PC XT/AT, Modula-2
Personal Consultant Easy (Texas Instruments)	Правила	Обратный вывод	PC XT, C, Scheme
Personal Consultant Plus (Texas Instruments)	Правила, фреймы, метаправила, факторы уверенности для нечетких знаний	Прямой и обратный выводы	PC XT/AT, C, Scheme
Procedural Consultant (Texas Instruments)	Деревья решений	—	PC XT, C, Scheme
Rule Master (Radian Corp.)	Деревья решений, правила	Конечные автоматы, индукция	PC XT, C
Savoir (ISI)	Нечеткая логика, формулы, демоны	Прямой и обратный выводы	PC XT, Pascal
Super Expert (Intelligent Terminals)	Множество примеров и деревьев решений	Прямой и обратный выводы, индукция	PC XT, Forth
VP Expert (Paperback Software)	Множество примеров, правила	Обратный вывод, индукция	PC XT, C
Xi Plus (Expertech)	Правила, отношения, факты, демоны	Прямой и обратный выводы	PC XT/AT, MicroProlog, assembler

Проблемно-ориентированные ИС

Название (фирма)	Область применения	Аппаратура, язык реализации
Expert Controller (Umecorp.)	Разработка программ и аппаратуры для управления процессами	PC XT
I-CAT (IN-ATE Lisp/Micro) (Automated Reasoning Corp.)	Диагностика неисправностей электронного оборудования	PC AT, Lisp
Interlliform (Mind Path Technologies)	Создание отчетов в реальном масштабе времени	PC XT, C
ProGenesis (Quantum inKNOWvations)	Интерфейс БЗ с реляционными СУБД, использующими язык SQL	PC XT, C
Pro. MD (Brainware)	Медицинская диагностика на основе технологии Mycin с использованием естественно-языкового интерфейса	PC XT, Prolog

Средние ИС для IBM PC и рабочих станций позволяют создавать более сложные ЭС (250...1000 правил). Для работы с такими ИС необходимы квалифицированные программисты и инженеры по знаниям. Средние ИС включают Personal Consultant Plus [23], GURU [12], Gold Works, Nexpert.

Перечисленные в табл. 1 ИС относятся к средствам универсального назначения для широкого спектра задач диагностики, планирования и процессов принятия решений. Наряду с ними все большее распространение приобретают проблемно-ориентированные (domain-specific) ИС [21, 25], предназначенные для разработки ЭС в достаточно узкой предметной области (табл. 2).

Представление знаний и способы вывода в ИС

Особенности представления знаний и стратегия логического вывода определяют два основных элемента ЭС — БЗ и машину логического вывода. Моделями (формализмами) представления знаний в ИС служат правила, фреймы, логика предикатов и смешанные подходы.

Большинство рассматриваемых ИС относится к системам, в которых для представления знаний в основном используются правила продукции вида «ЕСЛИ (условие), ТО (следствие)». Если левая часть правила (посылка) истинная в принятом в системе смысле (истинности некоторого предиката, наличие определенной информации в БД или БЗ и т. п.), то можно применить последовательность действий, указанных в правой части (следствии). При этом в продукционных системах процедуры анализа данных, выполняемые с помощью левой части, отделены от процедур модификации, осуществляемых правой частью. В целом для продукционных ИС характерно четкое разделение между данными, действиями и управлением [6].

В составе продукционных систем выделяют базу правил или БЗ, содержащую правила продукции; рабочую память или БД, отображающую текущее состояние решения задачи; управляющую структуру — интерпретатор правил или машину вывода, определяющую последовательность активизации правил.

Правила продукции устанавливают набор разрешенных преобразований, в соответствии с которыми происходит продвижение от начального состояния к окончательному решению задачи. Промежуточное состояние отображается

с помощью фактов, фиксируемых в рабочей памяти. В ходе решения (консультации) части правила в зависимости от используемого способа вывода сопоставляются с содержимым рабочей памяти — процедура сопоставления с образцом. Конфликтная ситуация, связанная с наличием нескольких правил, доступных для применения, разрешается интерпретатором правил. Действие правила заключается в изменении рабочей памяти.

Функции интерпретатора правил могут быть представлены как последовательность повторяющихся этапов:

- выбора правил и элементов данных, относящихся к текущему состоянию;
- сопоставления — отбора правил, для которых соблюдаются условия выполнения;
- установления очередности выполнения правил;
- выполнения.

Процедуры поиска решения, заложенные в интерпретаторе правил ИС, во многом определяют возможности ЭС и сферу их применения, поскольку они переносятся в разрабатываемую ЭС. Логический вывод — центральное звено нахождения решения в ЭС. Он означает построение заключения (заключительного высказывания) из цепочки высказываний (утверждений), истинных в данной ситуации. Такой вывод с булевыми значениями истинности назовем достоверным. Однако наиболее часто в ЭС используются так называемые правдоподобные рассуждения, при которых продукционные правила дают не вывод, а аргументацию (обоснование). При этом цепочки высказываний, реализуемые после выбора всех альтернативных продукций во всех конфликтных множествах, не эквивалентны в отличие от цепочек при достоверном выводе.

Во многих продукционных ИС в той или иной форме используются идеи, положенные в основу машин вывода Mycin, EMycin, Prospector [1, 7, 16]. Продукционные ИС наиболее часто реализуют прямой и (или) обратный выводы.

Обратный вывод, или вывод, управляемый следствиями (целями) правил, наиболее часто используется в ЭС для задач диагностики и интерпретации. При этом способе вывода происходит движение от следствий к посылкам. Вывод идет по всем альтернативным путям, которые отсекаются, если определенные посылки не выполняются. Процесс продолжается до тех пор, пока не будет найдено решение или установлено его отсутствие. Примером продукционных ИС с обратным выводом могут служить Crystal, Expert Edge, Exsys [24], VP Expert (см. табл. 1).

ЭС, создаваемые для решения задач синтеза (планирования, проектирования, выбора конфигурации), часто используют прямой вывод, или вывод, управляемый посылками. При прямом выводе просматриваются посылки и выявляются правила, посылки которых принимают истинное значение. Затем выполняются действия, связанные с этим правилом. Поскольку состояние рабочей памяти может измениться, процесс поиска правил с истинными посылками возобновляется.

В ИС Expertkit II, реализующей процедуру прямого вывода, правила можно объединить в пакеты и структурировать базу правил, выделяя правила, описывающие отдельную подзадачу.

Ряд продукционных ИС позволяет при разработке ЭС использовать как прямую, так и обратную процедуры вывода, применять их попеременно в рамках одного приложения, когда необходим многошаговый процесс поиска решения, поскольку заключительное состояние не достигается за один шаг.

К ИС, предоставляющим пользователям возможность выбора процедуры вывода, относятся GURU, Xi Plus Nexus, Savoig. В GURU разработчик может воспользоваться 50 различными вариантами процедур выбора правил для активизации. Интерпретатор правил настраивается с помощью специальных системных переменных, доступных разработчику.

Наряду с продукциями в ИС Xi Plus, Savoig допускается применение демонов — процедур, которые активи-

зируются при выполнении определенных условий в ходе консультации, в частности при изменении или вводе новых данных. В Xi Plus демоны имеют вид «КОГДА <условие>, ТО <действие>». По существу демоны — разновидность правил с тем отличием, что активизируются они сразу после выполнения условий, записанных в левой части. В ходе консультации демоны используются для реакции на события, предусмотренные разработчиком ЭС, например отображение на дисплее значений изменившихся переменных.

Эффективность ЭС можно существенно повысить, если разработчик получит возможность использовать метазнания — знания о характерных особенностях знаний конкретной предметной области, учет которых ускоряет процесс консультации. В Personal Consultant Plus таким средством являются метаправила, т. е. правила о применении обычных правил, позволяющие вводить в ЭС знания о способах решения задач и изменения приоритетов активизации правил в ходе консультации. Метаправила могут содержать дополнительную управляющую информацию, сокращающую пространство поиска решений в задачах большой размерности.

Продукционные ИС отличает простота и модульность представления знаний, а также независимость БЗ от структуры управления. Однако по большому количеству правил их организация в виде неструктурированного набора вызывает затруднения при внесении изменений в БЗ. Кроме того, не все знания экспертов достаточно удобно представляются с помощью формализма правил продукций.

Другой разновидностью систем, основанных на правилах, являются Super Expert [18], Micro Expert [5], Extran 7.2, использующие в качестве моделей представления знаний иерархию правил и дерева принятия решений.

Кроме того, выделяют ИС, основанные на смешанном фрейм-продукционном подходе: Leonardo [26], Personal Consultant Plus, Copernicus (S. 1), GoldWorks. Использование концепции фреймов — информационных структур для представления знаний-стереотипов об объектах внешнего мира с их атрибутами и взаимосвязями позволяет хранить знания о предметной области в структурированном виде, представлять в БЗ абстракции и аналогии. Правила и процедуры, относящиеся к определенным объектам, могут ассоциироваться с фреймами, что позволяет достаточно удобно структурировать БЗ. Система фреймов формирует структуры объектов, наследующие определенные атрибуты иерархии.

В Personal Consultant Plus фреймы используются для группирования фактов и правил, связанных с конкретными объектами, в частности система может содержать только один корневой фрейм. В процессе консультации фрейм активизируется, если подзадача (переменная, правило), описанная в нем, становится актуальной. Дерево фреймов подразделяет область экспертизы на подобласти. В Personal Consultant Plus, Leonardo фрейм-сын наследует параметры и правила от фрейма-отца.

Представляет несомненный интерес использование в различных ИС компонент приобретения знаний на основе данных, вводимых экспертом. В ИС Extran 7.2, Kiss, Rule Master [11], Super Expert, VP Expert, Xi Plus (Xi Rule) для генерации правил используется метод индуктивного вывода. В отличие от ИС, где правила в готовом виде вводятся в БЗ разработчиком, в индуктивных ИС пользователь вначале определяет набор первичных признаков и классы объектов. Затем создается множество обучающих примеров, содержащих значения атрибутов и результирующий класс, соответствующий данному множеству. На основе этих данных выводится (индуцируется) правило в виде дерева решений. Некоторые ИС генерируют программы на процедурных языках с представлением логики индукцированного дерева решений в виде вложенных выражений типа IF—THEN—ELSE, используемых в задачах классификации [25].

Одно из самых популярных ИС VP Expert сочетает в себе продукционную (дедуктивную) систему и индуктивную компоненту, позволяющую быстро изготовить прототип.

Средства работы в условиях неопределенности

Многие области применения ЭС характеризуются наличием в знаниях и оценках экспертов существенной доли неопределенности [27, 28]. Тем не менее такие знания играют важную роль в процессе экспертизы [5]: использование лишь беспорных фактов привело бы к отказу от решения большинством заинтересованных задач.

Для представления знаний и логического вывода в условиях неопределенности в коммерческих ИС в основном применяется байесовский подход, факторы (коэффициенты) уверенности (certainty factors) и нечеткая логика (fuzzy logic). При байесовском подходе вычисляется относительное правдоподобие конкурирующих гипотез путем взвешивания различных свидетельств с учетом неопределенности, заключенной в ответе пользователей. В некоторых ИС — потомках ЭС Prospector, например Micro Expert [5], суждениям экспертов присваиваются числовые значения. По шкале от -5 (нет, абсолютная ложь) до +5 (да, абсолютная истина). Байесовский подход применяется также в Crystal, VP Expert.

Однако в большинстве ИС (GURU, Exsys, Personal Consultant Plus) для указания степени достоверности знаний эксперта предпочтение отдается факторам уверенности (ФУ), впервые использованным Шортлифом в ЭС Mycin [15] для измерения степени доверия к любому заключению, являющемуся результатом полученных к данному моменту свидетельств. ФУ — это числа, оценивающие силу конкретного эвристического правила. Результирующие ФУ в ходе вывода решения, например в ИС, определяются в соответствии со следующими схемами:

устанавливается общий ФУ для посылки;
объединяются ФУ значений отдельной переменной, когда несколько правил выдали это значение с различными ФУ;
объединяются ФУ посылки правила и следствия для получения общего ФУ (16 различных схем). Конкретная процедура расчета выбирается разработчиком с помощью специальных системных переменных.

Байесовский подход и ФУ относятся к количественным методам работы в условиях неопределенности, а нечеткая логика, разработанная Заде [27] для выражения неопределенности лингвистических описаний предметной области, — к качественным.

Ряд ИС предлагает пользователю несколько способов работы в условиях неопределенности, например, в ИС Leonardo используются байесовская процедура, ФУ и нечеткая логика.

Несколько иной способ работы с нечеткими знаниями предлагается пользователям в ИС Xi Plus, основанном на символьном выводе, т. е. работе с описательными фразами на стилизованном английском языке. Пользователь может сформулировать любые бинарные отношения и утверждения, в содержании которых будут присутствовать слова, отражающие степень уверенности в знаниях ЭС. По мнению разработчиков системы, такой способ представления и манипулирования с нечеткими знаниями более привычен человеку и позволяет отказаться от ранее упомянутых искусственных статистических схем.

При создании БЗ Xi Plus (отношения, утверждения, правила, демоны) разработчик может использовать наряду с ключевыми любыми английскими словами, а также фразы на русском языке.

Интерфейс пользователя

Создание интерфейса пользователя занимает ~50 % времени разработки. В коммерческих ИС на ПЭВМ пока не нашли широкого применения средства понимания естественного языка. В то же время использование ключевых слов английского языка для разработки интерфейса на ограниченном естественном языке доступно в GURU. Такой интерфейс удобен для тех, кто хочет получить консультацию у ЭС, не изучая специальных терминов. GURU имеет встроенный словарь объемом около 500 слов. Пользователь может вводить по мере необходимости «Микропроцессорные средства и системы» № 6, 1989 59

димости новые слова. В ЭС GURU, Leonardo, Personal Consultant Plus консультации сопровождаются выдачей на экран графической информации.

Один из важнейших элементов ЭС — компонента объяснения. Чтобы помочь эксперту и разработчику совершенствовать ЭС, система должна обладать способностью объяснить ту цепочку рассуждений, которая привела к определенному состоянию консультации. Пользователь в этом случае может проанализировать ход консультации в терминах целей и правил.

Большинство рассматриваемых ИС позволяют встраивать в ЭС ретроспективный механизм выдачи объяснений, обеспечивающий запись процесса вывода. Объяснения рассматриваются в терминах движения от текущей цели по дереву целей.

Пользователь в процессе консультации может обратиться с вопросом: КАК (HOW) было получено конкретное заключение? В ответ ему выдается вся цепочка рассуждений с указанием задействованных правил, т. е. осуществляется пошаговое восстановление истории вывода. На вопрос ПОЧЕМУ (WHY) ЭС выдает обоснование результата в виде «основных соображений».

Ответы формируются как результат анализа цепочки вывода и извлечения поясняющих текстов, которые на этапе создания ЭС были отнесены к правилам или фреймам БЗ. В Xi Plus объяснения системы могут выдаваться в графической форме и отображаться в специальном окне на дисплее.

Интересной особенностью Xi Plus является возможность изучения последствий воздействия одного или нескольких ответов пользователя ЭС на процесс вывода заключений. В любой момент диалога на вопрос ЧТО, ЕСЛИ (WHAT IF) ЭС выводит на экран перечень всех вопросов, заданных пользователю, и соответствующие ответы. При этом ответы на некоторые вопросы можно изменить для изучения альтернативных путей вывода или исправления ошибок.

Ряд ИС, в частности Xi Plus, предоставляет возможность ввода всей имеющейся у пользователя информации, не дожидаясь вопросов ЭС.

Интерфейс разработчика

ИС обеспечивают, как правило, два режима работы — разработки и консультации. В режиме разработки создается, изменяется, отлаживается БЗ, инициируется режим консультации. В GURU, Xi Plus, Personal Consultant Plus и некоторых других ИС разработчик может взаимодействовать с системой на уровнях меню и командного языка. Используемые в этом режиме редакторы БЗ и экранных форм, генераторы отчетов и разнообразные отладочные средства конечному пользователю не предоставляются.

Таким образом, ИС представляет собой совокупность компонент, обеспечивающих функционирование ЭС (средства ввода, хранения и манипулирования знаниями в БЗ, машины вывода, объяснительная компонента), непосредственно входящих в состав готовой ЭС и используемых только в ходе разработки ЭС.

Наиболее простые и дешевые ИС Micro Expert для создания БЗ содержат обычные текстовые редакторы, с помощью которых разработчик формализует знания экспертов на языке представления знаний конкретной ИС. Затем БЗ компилируется для выявления ошибок и компактного представления в памяти ПЭВМ.

В более совершенных ИС предусмотрены специализированные редакторы БЗ, компилирующие знания из формы, представленной на ограниченном естественном языке, в форму на языке представления знаний ИС и проверяющие заносимый в БЗ элемент знаний. Наиболее простые из них осуществляют лишь грамматический разбор высказываний экспертов и сообщают об ошибках. Более сложные формируют готовые структуры БЗ в ходе диалога с пользователем, выявляют конфликты между уже хранящимися и

вновь вводимыми знаниями и проверяют их семантическую правильность. Разработчик ИС имеет возможность в ходе диалога устранить ошибки, ввести ранее неопределенные понятия и вывить заикливание или разрыв в цепочке рассуждений ЭС.

Отладочные средства включают трассировку, определяющую последовательность действий ЭС, активизации правил и фреймов, вызов функций и т. д.; механизм прерываний, приостанавливающий исполнение программы, изменяющий и анализирующий состояние переменных при возникновении ошибок.

Например, отладчик Xi Plus может быть запущен в любой момент разработки БЗ. Результатом его работы являются:

- список и значение всех идентификаторов и утверждений, используемых в БЗ;
- перекрестные ссылки имен, значений, правил и вопросов; изолированные или неиспользуемые правила;
- условия в правилах и значения в отсутствующих вопросах;
- если вопросы отсутствуют и в процессе консультации, то ИС сама генерирует соответствующий вопрос;
- указания на места возможного заикливания в ходе вывода;
- конфликтные значения идентификаторов.

В режиме разработки пользователь Xi Plus имеет доступ к командному языку, редактору и отладчику БЗ, редактору форм, средствам спецификации интерфейсов с внешними программами. Для каждой предметной области создается свое приложение, допускающее работу с несколькими БЗ, каждая из которых может характеризовать различные аспекты области экспертизы. ИС позволяет с помощью меню переключаться с одного приложения на другое. БЗ адресуется непосредственно или связываются в цепочку и вызываются по мере необходимости.

Простейший способ использования БЗ — активизация запроса. Возможные запросы включают разработчиком в БЗ на этапе ее создания. В режимах разработки и консультации пользователю Xi Plus доступна глобальная структура данных, содержащая все, что относится к текущему состоянию консультации, — активизированные правила, значения переменных, вопросы и т. д. Такая структура данных позволяет оперативно предоставлять пользователю исчерпывающие сведения о ходе экспертизы.

Интерфейс разработчика с ИС может включать несколько уровней в зависимости от квалификации пользователя. Например, Personal Consultant Plus для описания правил предоставляет два уровня: язык ARL для компактной записи правил на стилизованном английском языке; язык Лисп для прикладных программистов.

В развитых ИС значительное внимание уделяется вопросам санкционирования доступа конечных пользователей к БЗ ЭС, поэтому в GURU, Xi Plus имеются средства создания защищенных БЗ.

Системный интерфейс

При выборе ИС немаловажную роль играет наличие в нем доступа к внешним программам — коммерческим пакетам, ОС, программам пользователей, созданным на основе традиционной технологии программирования.

Разработчики ИС в первую очередь ориентируются на средства, обеспечивающие интерфейс ЭС с ОС MS DOS (PC DOC), в частности на обращение к функциям ОС с последующим возвратом в среду ЭС, а также на работу с файлами, представленными в стандарте ASCII (табл. 3).

Многие ИС не обеспечивают ЭС собственной базой данных, поэтому в них предлагаются интерфейсы с dBase II, III, III+, Lotus 1-2-3, Symphony и другими коммерческими пакетами. Наконец, для интегрирования ЭС с программами на процедурных языках, обеспечивается интерфейс с языками С и ассемблера. Как правило, ИС имеет интерфейс для языка, на котором оно реализовано.

Интерфейсы ИС с пакетами прикладных программ и системным программным обеспечением

ИС	Внешний интерфейс
Crystal	MS DOS, Lotus 1-2-3, dBase III, Symphony, C
ESP Advisor	Prolog-2
Expert Edge	dBase III, Lotus 1-2-3, VisiCalc, MS DOS
ESP Frame Engine	Prolog-2
Experkit II	LE Lisp
Exsys	dBase II
Kiss	Micro Prolog
M.1	C, assembler
Personal Consultant Plus	dBase II, dBase III
Savoir SuperExpert	Pascal, Fortran, dBase III, Videotex
VR Expert	Lotus 1-2-3, dBase III, dBase II, dBase III, dBase III+, Lotus 1-2-3, VP-Info, VP-Planner
Xi Plus	Assembler, C, GEM Draw, dBase, Lotus 1-2-3

Представляет интерес и создание интегральных ИС, объединяющих технологию ЭС с БД, электронными таблицами и пакетами прикладных программ. Такой подход применен в GURU, где наряду с ИС для разработки ЭС интегрированы: система управления БД — Knowledge Map, использующая язык SQL, электронная таблица, пакет деловой графики и средства создания псевдоестественно-языкового интерфейса.

Заключение

В отличие от ИС для больших и мини-ЭВМ, реализуемых на Лиспе и Прологе и языках процедурного типа, разработчики ИС для ПЭВМ ориентируются в основном на процедурные языки, как наиболее эффективные и быстродействующие. В частности, даже такое удачное ИС, как Xi Plus, реализованное на Микропрологе, будучи переписанным на язык С (Xi Plus, версия 3) позволяет повысить быстродействие в несколько раз. Однако появление специализированных Пролог- и Лисп-машин, сравнимых по цене с ПЭВМ, может существенно изменить соотношение сил в пользу ИС, базирующихся на языках искусственного интеллекта.

Приведенные в работе сведения по ИС для ПЭВМ не претендуют на полное представление о рынке коммерческих ИС, который в настоящее время быстро развивается и, по оценкам экспертов, является весьма перспективным.

Из отечественных ИС для ПЭВМ известны совместная советско-болгарская система Интерэксперт — функциональный аналог GURU, системы СПЭИС (ВНИИСИ АН СССР) [3] и ПИЭС (ВЦ АН СССР) [29]. С развитием аппаратных и программных средств ПЭВМ в направлении применения 32-разрядных микропроцессоров пользователям ПЭВМ окажутся доступными и ИС, которые в настоящее время применяются только на больших и мини-ЭВМ, — ART, KEE, KES-11, DUCK, Knowledge Craft и др., обладающие значительно большими функциональными возможностями.

Телефон 434-82-24, Москва

ЛИТЕРАТУРА

1. Построение экспертных систем: Пер. с англ. / Под ред. Ф. Хейеса-Рота, Д. Уотермана, Д. Лената. — М.: Мир, 1987. — 441 с.

2. Поспелов Г. С., Поспелов Д. А. Искусственный интеллект — прикладные системы // Математика и кибернетика. — 1985. — № 9. — С. 1—34.
3. Геловани В. А., Ковригин О. В. Экспертные системы в медицине // Математика и кибернетика. — 1987. — № 3. — С. 1—32.
4. Попов Э. В. Экспертные системы. — М.: Наука, 1987. — 288 с.
5. Экспертные системы: принципы работы и примеры. — М.: Радио и связь, 1987. — 223 с.
6. Элти Дж., Кумбс М. Экспертные системы: концепции и примеры. — М.: Радио и связь, 1987. — 191 с.
7. Алексеева Е. Ф., Стефанюк В. Л. Экспертные системы — состояние и перспективы // Известия АН СССР. Сер. Техническая кибернетика. — 1984. — № 5. — С. 153—167.
8. Микулич Л. И. Проблемы создания экспертных систем // Теория и модели знаний. — 1985. — № 714.
9. Waterman D. A. A Guide to Expert Systems. — Addison-Wesley, 1986. — 456 p.
10. Buchanan B. G. Expert systems: working systems and the research literature // Expert Systems. — 1986. — V. 3, N 1.
11. Lehner P. E., Barth S. W. Expert Systems on microcomputers // Expert Systems. — 1985. — V. 2, N 4.
12. Shafer D. G. Microcomputer-based Expert Systems: Where we are headed // Expert Systems. — 1985. — V. 2, N 4.
13. Электроника. — 1986. — Т. 59, № 16 — С. 18—31.
14. Cohen F., Guiltyle C. Explore Smalltalk and OPS // Expert System User. — 1987. — V. 12, N 2.
15. Shortliffe E. N. Computer-based Medical Consultation MYCIN. — Elsevier, New-York, 1976.
16. Catalog of Artificial Intelligence Tools. / Ed. A. Bundy / — Springer-Verlag, 1986. — 168 p.
17. Cohen F. Listen to a selection of European Shells // Expert Systems User. — 1987. — V. 12, N 12.
18. Liardet M. Good Breeding Shows in ITL's Super Expert // Expert Systems User. — 1987. — V. 12, N 11.
19. New version of M1 // Expert Systems. — 1985. — V. 2, N 4.
20. Helix develop new shell // Expert Systems. — 1985. — V. 2, N 2.
21. Currently Available Expert System Building Tools // Expert Systems Strategies. — 1988. — V. 4, N 7.
22. Crystal // Expert Systems Strategies. — 1988. — V. 4, N 10.
23. Enhanced Personal Consultant from TI // Expert Systems. — 1985. — V. 2, N 4.
24. Exsys: shell for PCs // Expert Systems. — 1985. — V. 2, N 4.
25. Two New Tools // Expert Systems Strategies. — 1988. — V. 4, N 4.
26. Roth A. Bridging the Gap? // Expert Systems User. — 1988. — V. 13, N 12.
27. Поспелов Д. А. Ситуационное управление. — М.: Наука, 1986. — 218 с.
28. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенного решения. — М.: Мир, 1976. — 165 с.
29. Хорошевский В. Ф. Разработка и реализация экспертных систем — инструментальный подход // Известия АН СССР. Сер. Техническая кибернетика. — 1986. — № 5. — С. 104—105.

Статья поступила 23.02.89

ПОСТРОЕНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ СРЕДСТВАМИ dBASE

Предлагается пример практической реализации интеграции средств СУБД и экспертных систем (ЭС): решение задачи классификации или выбора среди фиксированного набора альтернатив на основе информации, вводимой пользователем, т. е. задачи создания ЭС, помогающей пользователю выбрать из всего многообразия однотипных программных средств те, которые ему больше всего подходят.

Существует несколько систем такого типа. На примере конкретной системы рассмотрим возможности использования средств dBASE II и DBASE III для решения подобных задач. При создании описываемой системы автор исходил из следующих допущений:

любая экспертная система в процессе работы манипулирует тем или иным набором данных;

некоторая часть знаний в принципе неотличима от данных, по крайней мере на стадии хранения;

для доступа к данным и знаниям целесообразно применять специально для этого предназначенный инструментари в виде СУБД.

СУБД с развитыми средствами доступа к данным и организации диалога пользователя позволяет построить все подсистемы ЭС средствами этой СУБД. В частности, программные средства dBASE обладают достаточной гибкостью, допускают рекурсивное обращение к процедурам и развитым аппаратом булевой логики. Это позволяет создавать все основные подсистемы ЭС, в том числе машину вывода и подсистему объяснения хода консультации.

В процессе формирования ЭС можно выделить два крупных этапа: разработку и отладку демонстрационного прототипа ЭС и изготовление рабочей версии ЭС, предоставляемой пользователю. Принято считать, что на первом этапе разработки наиболее рационально использовать «оболочки» или «пустые ЭС», а после отладки, тестирования и т. д., т. е. после завершения этого этапа, переписать систему на какой-либо алгоритмический язык для увеличения быстродействия, экономии памяти и обеспечения переносимости на другие ЭВМ. Так, широко известные системы MYCIN, DENDRAL, HAL, INTERNIST, SECS написаны на языках INTERLISP, LISP, BASIC, FORTRAN, SNOBOL. Этот список можно продолжить.

На первом этапе разработки системы использование языка СУБД dBASE III не многим сложнее, чем какой-либо «оболочки» или систем типа KEE, основанных на фреймах, или типа ART — на правилах и т. п. Более того, построение подсистем доступа к базе дан-

ных, диалогового общения и объяснения процесса принятия решения, по-видимому, проще выполнить именно этими средствами. Большая гибкость языка СУБД (как, впрочем, и любого алгоритмического) по сравнению со средствами, предоставляемыми простыми «оболочками», позволяет организовать более разнообразные структуры интерпретатора правил. Кроме того, если не предъявляется каких-либо специальных требований к быстродействию системы, можно вообще отказаться от второго этапа разработки и использовать систему в том виде, в каком она получилась на первом этапе работы.

При разработке ЭС в описываемой задаче использовалось представление правил в виде ЕСЛИ ... ТО ... Но упомянем и некоторые другие возможности. Так (без претензии на абсолютную корректность), нетрудно усмотреть следующие параллели между фрейм-структурами и файлами dBASE:

Фрейм	dBASE
Фрейм-прототип	Структура
Слот	Поле записи
Фрейм-экземпляр	Экземпляр записи

В dBASE III реализуется и процедурное представление знаний, например следующим образом: в поле name 1 записывается левая часть правила; в поле name 2 — правая часть правила в виде file 1, где file 1 — имя некоторого командного файла. Соответствующее место программы выглядит так:

```
IF name1 = a
  STORE name2 TO m
  DO &m
ENDIF
```

Запуск командного файла можно поручить пользователю как один из альтернативных вариантов продолжения работы с системой. В этом случае программа примет вид

```
IF name1 = a
  STORE name2 TO m
  @ 10,10 SAY 'Выполнено условие 1'
  @ 12,10 SAY 'для продолжения нажмите'
  @ 14,10 SAY 'клавиши &m'
  WAIT
ENDIF
```

Кроме того, можно, конечно, использовать пирамиду «ноу-хау», иначе именуемую иерархией знаний, но следует иметь в виду, что это затрудняет расширение базы знаний.

Описание программы

Программа решает задачу анализа, или, более конкретно, задачу классификации и оценки программных пакетов. В состав системы входят: база данных, подсистема диалогового общения, подсистема объяснения, решающий блок

и справочная подсистема, позволяющая получить полную информацию о любом из включенных в систему пакетов.

База данных содержит характеристики 27 текстовых редакторов (по 100 характеристик для каждого).

Система использует знания в виде правил. Типичное правило в записи на русском языке выглядит так:

ЕСЛИ

Цена пакета больше заданной,

ИЛИ

Требуемая память больше заданной.

ИЛИ

Требуемое число дисководов больше заданного.

ТО

Изменить коэффициент доверия к пакету.

Правила выбора пакетов записываются в два обычных файла данных dBASE. Структура первого содержит три поля: номер правила, текст левой части правила и имя поля базы данных, подлежащего обработке. Второй файл — значения левой части правила и коэффициенты доверия. Его запись содержит поля: номер правила, разрешенное значение левой части, коэффициент доверия, примечание.

Сеанс работы с системой в основном состоит в том, что пользователь на последовательности предлагаемых ему меню отмечает существенные для него характеристики пакетов. Система анализирует ответы, отбрасывает пакеты, не обладающие заданными характеристиками, оставшиеся пакеты оценивает в соответствии с заложенными в ней правилами и выдает пользователю перечень отобранных пакетов и их оценок.

Типичные подсистемы диалогового общения:

Подходящие операционные системы:

1. MS DOS
2. CP/M
3. OASYS
4. RDOS
5. MP/M
6. APPLE DOS

Выход:

Максимальная цена пакета (измените или оставьте)	— 695 руб.
Минимальное число дисководов (измените или оставьте)	— 2.
Минимальная память для пакета (измените или оставьте)	— 256 Кбайт.

На первом экране пользователю предлагается выбрать одно или несколько из разрешенных значений левой части правила. В первой строке печатается текст левой части, дальше выводится список разрешенных значений. Пользователь выбирает значения, переводя на них «святиющую строку

(с помощью клавиш-стрелок) и нажимаемая клавиша <Ввод>. Затем система анализирует содержимое поля базы данных, указанного в файле правил, и меняет значение коэффициентов доверия (КД) в соответствии с ответом пользователя. На втором экране пользователю предлагается ввести удовлетворяющие его числовые значения, и система сравнивает эти значения с содержимым базы данных.

Значения КД, приписываемые правилу, могут меняться от 0 до 100. После применения очередного правила новое значение вычисляется по формуле $KD = (KD1 \cdot 100 + KD2 \cdot 100 - KD1 \times KD2) / 100$.

Время обработки ответов пользователя на каждом шаге (при обработке ответов на каждое меню) составляет от долей секунд до нескольких секунд. Полное время, которое система затрачивает на анализ (полный сеанс работы с системой), составляет несколько минут. Это время уменьшается, если какие-либо группы характеристик пользователь отмечает как несущественные. Вся система помещается на одной дискете и занимает около 250 Кбайт.

Важная характеристика любой ЭС — ее способность к расширению. В данном случае табличная организация базы данных позволяет очень просто ее наращивать. Кроме того, упрощается запись правил, так как в файл правил достаточно один раз ввести левую часть правила, а затем перечислить ее разрешенные значения. Уменьшается вероятность ошибок при ответе пользователя: ему не нужно вводить с клавиатуры текст ответа (разрешенное зна-

чение выбирается перемещением световой строки).

«Классические» подсистемы ЭС. В нашей задаче применялась последовательная проверка правил, при которой на основании ответов пользователя и фактов, содержащихся в базе данных, оцениваются коэффициенты доверия различных гипотез. Так как язык команд dBASE допускает рекурсивное обращение к процедурам и обладает широкими возможностями использования булевой логики, средствами dBASE можно создать и машину вывода с «прямой» или «обратной» цепочкой правил.

Большинство современных ЭС включают в себя подсистему объяснения хода консультации. При последовательном переборе правил такая подсистема предоставляет пользователю информацию о текущем правиле, а при выводе прямой или обратной цепочки на экран выводятся тексты всех правил, цепочки с соответствующими КД.

Выводы

В связи с быстрым ростом популярности ЭС увеличивается потребность в небольших ЭС, рассчитанных на ограниченный круг пользователей. Изготовление таких систем традиционными методами с помощью «оболочек» или символьных языков типа ЛИСП и ПРОЛОГ, по-видимому, не всегда оправдано. Во-первых, все эти средства не являются универсальными, например в отношении представления знаний, и для разных задач приходится выбирать различные средства. Кстати говоря, до сих пор нет единого мнения о том, нужна ли и возможна ли такая универсальная система. Во-вторых, эти

средства недешевы и не всегда легко доступны. Кроме того, для них требуются дорогостоящие ПЭВМ с большим объемом оперативной памяти. Все это привело к разработке многочисленных ЭС, написанных на обычных языках программирования, таких как Фортран, Паскаль, БЕЙСИК и т. п. Предпринятая попытка создания ЭС средствами СУБД, на наш взгляд, подтверждает, что таким способом можно достаточно быстро создавать недорогие и удобные ЭС для ПЭВМ типа Роботрон 1715 или ЕС1840. При не слишком большом числе правил (около 100) они позволяют легко наращивать и изменять базу данных системы и подсистему принятия решений, что повышает адаптивность системы к изменяющимся условиям.

Телефон 255-06-98, Москва

ЛИТЕРАТУРА

1. Al-Zobaidie A., Grimson J. B. Expert systems and database systems: how can they serve each other? // Expert Systems.— 1987.— Vol. 4, N 1.— P. 30—43.
2. Integrating AI & database technologies. Topics in information systems. On knowledge base management systems.— Springer-Verlag, 1986.
3. Vassiliou Y. Integrating database management and expert systems // Informatik-Fachberichte.— 1985.— N94.— P. 147—160.
4. Hession W., Rubel M. Performance guide to word processing software.— McGraw-Hill, 1985.— 215 p.

Статья поступила 9.12.87.

УДК 681.3.06

У. Ф. Фейзханов

ПОДПРОГРАММЫ ВВОДА-ВЫВОДА БОЛЬШИХ МАССИВОВ ДАННЫХ ДЛЯ ДВКЗ

При неформатном вводе-выводе данных в ДВК, т. е. вводе-выводе без преобразования и поэтому самым быстрым, максимальная скорость обмена с фиктивным устройством (самым быстрым устройством обмена в ОС РТ-11) составляет всего 9 Кбайт/с. Это связано с тем, что элементарной единицей чтения-записи ОС РТ-11 служит физический блок, равный 512 байт, а запись данных при прямом доступе в Фортране происходит порциями по 4, 8, 12, ... байт. Поэтому в средствах Фортрана предусмотрена буферизация операции обмена: данные записываются сначала в буфер, из которого по мере заполнения переписываются в файл. При обработке больших массивов данных обмен происходит не отдельными элементами массива, а блоками данных размера 256, 512, 1024, ... $2 \times N$ слов. В этом случае целесообразно реализовать операции ввода-вывода без буферизации и существенно ускорить обмен.

Для такого обмена предлагается следующий набор подпрограмм, написанных на Макроассемблере:

OPW (NCH, NAME, ISIZE, IBLOK) — открывает новый файл;
 OPR (NCH, NAME, ISIZE, IBLOK) — открывает существующий файл;
 WR (NCH, MAS, IZAP), RD (NCH, MAS, IZAP) — запись и чтение одного блока данных;
 CLS (NCH) — закрывает файл и освобождает канал (одновременно могут быть открыты 15 каналов);
 NAME — имя файла, в кодах ASCII;
 ISIZE — размер блока данных в словах (наиболее эффективно использовать размер 256, 512, 768, ..., $256 \times N$);
 IBLOK — число блоков в файле;
 MAS — массив данных в оперативной памяти;
 IZAP — номер блока в файле.
 Скорость обмена программы пользователя с фиктивным устройством с помощью этих подпрограмм составляет 3,5 Мбайт/с, емкость занимаемой памяти — менее 1 Кбайт.

Телефон 535-08-06, Москва

Сообщение поступило 28.02.89

УДК 681.3

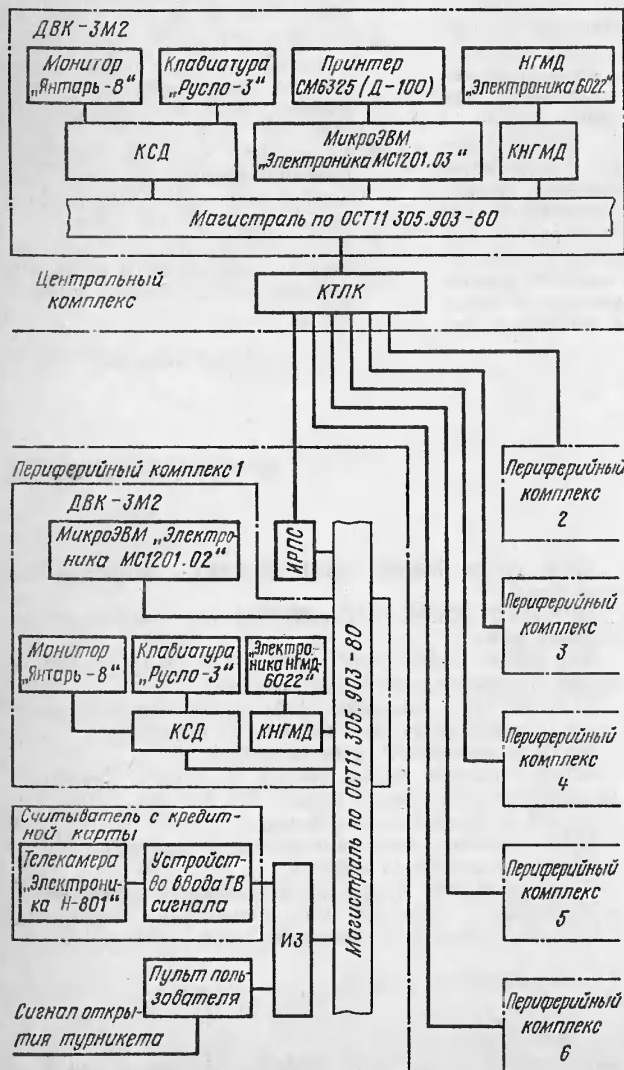
В. Г. Коломыц, И. В. Королев, А. Д. Чернявский

СИСТЕМА АВТОМАТИЗИРОВАННОГО КАССОВОГО РАСЧЕТА

Технические средства системы

Комплекс технических средств системы автоматизированного кассового расчета (САКР) выполнен на базе серийно выпускаемых диалоговых вычислительных комплексов ДВКЗМ2 с использованием дополнительных контроллеров периферийных устройств для микроЭВМ семейства «Электроника» и специально разработанного оборудования. Структурная схема САКР (см. рисунок) состоит из центрального и периферийного комплексов.

Центральный комплекс системы выполнен на базе ДВКЗМ2, в котором установлена одноплата микроЭВМ



«Электроника МС1201.02» («Электроника НЦ8001 ДА») и плата КТЛК для обмена информацией по последовательному каналу связи с периферийными комплексами. Назначение центрального комплекса: загрузка периферийных комплексов всей необходимой информацией (меню, разрешенные кредитные номера); разрешение работы периферийного комплекса в том или ином режиме; прием, накопление и хранение информации об оплаченных обедах; предварительное формирование носителя для передачи информации об оплаченных обедах на ЕС ЭВМ; формирование документов и отчетов о работе столовой; решение вспомогательных задач.

Через плату КТЛК по интерфейсу ИРПС к центральному комплексу подключается до шести периферийных комплексов. Назначение периферийного комплекса: отображение меню на дисплее; ведение диалога с клиентом; распознавание кредитного номера и допуск клиента в зону раздачи; передача в центральный комплекс кредитного номера клиента, выбранных к оплате блюд и их стоимости.

Периферийный комплекс выполнен на базе ДВКЗМ2. К линии связи он подключается через блок ИРПС, используемый для аналогичных целей в БК-0010Ш [1]. Для упрощения диалога и считывания кредитного номера разработаны дополнительные периферийные устройства: пульт пользователя, считыватель с кредитной карты. Они позволяют использовать клавиатуру ДВК только во время загрузки рабочей программы и автоматизировать ввод кредитного номера клиента.

Считыватель с кредитной карты вводит в ЭВМ кредитный номер клиента, представленный штриховым кодом. К каналу микроЭВМ «Электроника МС1201.02» считыватель подключен через интерфейс прямого доступа к памяти ИЗ (15КС-16-002). Считыватель состоит из бытовой видеокамеры «Электроника Н-801» и устройства ввода телевизионного сигнала. Работа микроЭВМ с устройством ввода телевизионного сигнала производится через четыре программно-доступных регистра интерфейса прямого доступа к памяти ИЗ:

регистр счета слов (РСС)	(адрес 172410);
регистр адреса (РА)	(адрес 172412);
регистр состояния (РС)	(адрес 172414);
регистр выходных данных (РВД)	(адрес 172416).

Пульт пользователя также подключен к ЭВМ через интерфейс ИЗ и предназначен для упрощения диалога клиента столовой с ЭВМ и сокращения времени на обслуживание в режиме произвольного выбора и расчета за обеды. Пульт управляет состоянием определенных разрядов РС состояния интерфейса прямого доступа к памяти, которые затем считываются микроЭВМ. Устанавливая с помощью коммутационных кнопок в единичное состояние разряды 09, 10, 11 (СТ1, СТ2, СТ3) РС Из3 (адрес РС 172414), можно перемещать указатель блюд на экране терминала, выбирать блюдо или отменять выбор, идентифицировать окончание расчета. Запись со стороны микроЭВМ единицы в разряд 02 (РУ2) зажигает индикатор на пульте пользователя и формирует сигнал на открытие турникета. На лицевой панели пульта размещены кнопки: выбора и отмены блюд, перемещения указателя выбора блюд, окончательного расчета за обед, индикатор включения турникета и разрешения прохода в зону раздачи.

Программное обеспечение системы

Программное обеспечение (ПО) САКР столовой предприятия обеспечивает работу всего комплекса технических средств. Отдельно разработаны ПО центрального и периферийных комплексов. Функционирует ПО в операционных системах РАФОС, ОС ДВК.

Программное обеспечение центрального комплекса управляет работой периферийных комплексов и ведет статистику работы систем. Функции ПО: определение конфигурации системы и назначение режима работы периферийному комплексу (в диалоге с оператором); контроль наличия файлов, содержащих меню для каждой линии; начальная загрузка информации о кредитных номерах, разрешенных к использованию в системе расчета в кредит; передача меню в периферийные комплексы в случае изменения его на центральном комплексе; передача информации о кредитных номерах в периферийные комплексы; изменение меню для любой линии во время работы системы; сбор статистики о работе периферийных и центрального комплексов; ввод, контроль, учет кредитных номеров. В состав ПО центрального комплекса входят программы ввода контроля, коррекции кредитных номеров и обслуживания периферийных комплексов.

Программа ввода, контроля и коррекции кредитных номеров обеспечивает ввод, контроль и коррекцию 20 000 кредитных номеров. Учтывая, что вся информация о кредитном номере заключается в двух значениях (запрещен/разрешен), а также необходимость обеспечения к ней быстрого доступа, решено использовать для хранения всех кредитных номеров максимально плотную упаковку. В результате для одного кредитного номера отводится 1 бит памяти и для хранения информации о 20 000 номерах потребовалось 1250 (16-разрядных) слов. Программа позволяет производить следующие действия над множеством кредитных номеров: ввод разрешенных и запрещенных номеров в виде интервалов, отдельного запрещенного номера, отдельного разрешенного номера; вывод всех разрешенных номеров на печать, терминал или в файл; сохранение предыдущих данных в резервном файле.

Программа обслуживания периферийных комплексов и сбора статистики о работе системы предназначена для начальной загрузки периферийных комплексов всей необходимой информацией. Функции, выполняемые программой: определение конфигурации системы в диалоге с оператором; контроль наличия и коррекции файлов, содержащих меню для каждой линии; начальная загрузка информации о кредитных номерах, разрешенных к использованию в системе расчета в кредит; передача меню в периферийные комплексы; передача меню в периферийные комплексы после изменения; при необходимости изменение меню для любой линии во время работы системы; сбор статистики о работе периферийных и центрального комплексов.

Сбор статистики — одна из основных функций центрального комплекса — ведется по двум направлениям: учет всех меню и всех посетителей за день. Посетители учитываются следующим образом. После приема заказа у клиента на любом периферийном комплексе на устройство печати и в файле протокола выводится строка, содержащая следующие данные: номер линии, где был принят текущий заказ; информацию о виде меню — комплекс или свободный выбор, в последнем случае — номера выбранных блюд; вид расчета — кредит (кредитный номер клиента); общая сумма заказанного обеда; время принятия заказа.

Статистика о работе центрального комплекса включает в себя именованную систему, дату, время загрузки и полную информацию о меню, использовавшихся при работе системы. Как только какое-либо меню передается в одну из линий, оно записывается в файл протокола с указанием времени передачи и номера линии.

Программное обеспечение периферийного комплекса. Функции ПО: прием из центрального комплекса меню для данной линии и кредитных номеров, разрешенных для расчета в кредит; прием заказа на комплексный обед; считывание штрихового кода с кредитной карточки, распознавание, проверка на «разрешенность»; передача информации о сделанном заказе и кредитном номере в центральный комплекс; контроль правильности передачи информации с помощью подсчета и проверки контрольной суммы.

Программа работы периферийного комплекса загружает с накопителя на гибком магнитном диске «Электро-

ника НГМД-6022». Устройства внешней памяти «Электроника НГМД-6022» используются только для первоначальной загрузки программы. В дальнейшем предполагается программу работы периферийного комплекса зашить в ПЗУ, что позволит обеспечить загрузку программы по включению питания. Программная периферийного комплекса поддерживает работу считывателя штрихового кода с кредитной карты, пульта пользователя и дисплея.

Для совместной работы с линиями связи центрального и периферийного комплексов было разработано специальное ПО. Процедура обмена по линии связи использует следующие восьмеричные адреса приемников и передатчиков.

Для периферийных комплексов:

приемник
адрес регистра состояния (РС) 176560
адрес регистра данных (РД) 176562
передатчик
адрес РС 176564
адрес РД 176566

Для центрального комплекса (линия 1):

приемник
адрес РС 176560
адрес РД 176562
передатчик
адрес РС 176564
адрес РД 176566

Следующие линии имеют соответствующие адреса, увеличивающиеся последовательно на 10.

Программа, принимающая с линии сигнал, определяет его появление в РД приемника по шестому биту РС приемника, и если бит установлен в единицу, значит, в РД имеется символ, пришедший из линии. Используются два режима приема сигнала из линии:

1) постоянное слежение за РС приемника, при этом все сигналы, поступающие из линии, будут приняты. Режим используется для приема больших посылок (меню, кредитных номеров, информации о заказе);

2) наблюдение за РС приемника в течение произвольного промежутка времени. Таким образом, обеспечивается принятие сигналов, поступивших из линии в этот промежуток времени. Режим используется центральным комплексом для последовательного опроса линий, а периферийными комплексами — при обменах с центральными комплексами одиночными сигналами.

В линию сигнал посылается, как только в РС передатчика будет установлен в единицу шестой бит, сигнализирующий о готовности принять символ. Устанавливается задержка, необходимая для того, чтобы программа на другом конце линии успела обработать или переместить в буфер принятый символ. В противном случае может произойти переполнение РД приемника и все сигналы, поступившие после этого, будут потеряны.

Функционирование всей системы

После загрузки системы оператор на центральном комплексе определяет режимы работы периферийных комплексов — назначаются меню для каждой линии с возможностью использования одного меню несколькими линиями. Программа в центральном комплексе производит последовательный опрос активных линий, откуда поступают следующие сигналы: требования начальной загрузки, передачи меню, передачи кредитных номеров и принятий данных о сделанном заказе; подтверждения/отрицания совпадения контрольной суммы передачи посылки; требования остановить и продолжить передачу; готовности принять данные.

Получив сигнал о требовании начальной загрузки, программа в центральном комплексе передает в линию соответствующее меню (объемом от двух до четырех блоков по 512 байт) с подсчетом и проверкой контрольной суммы передаваемой информации. Далее передается информация о кредитных номерах (три блока по 512 байт), и после этого периферийный комплекс приступает к обслуживанию клиентов. Меню, переданное в периферийные комплексы,

фиксируется в файлах протоколов (с указанием времени передачи и номера линии). После начальной загрузки программа периодически с интервалом 4...5 с предоставляет оператору возможность выбрать следующие режимы работы: закончить работу системы, изменить меню для какой-либо линии. Если будет выбран режим изменения меню, то соответствующее меню выводится на экран оператора и он может с помощью функциональных клавиш подвести маркер по экрану к необходимому информационному полю и изменить его. Если произошло изменение меню, то при последующем обмене с соответствующей линией оно будет туда передано.

Работа периферийных комплексов. Заказы на периферийных комплексах принимаются в интерактивном диалоге с клиентом, от которого не требуется специальных знаний, используется только одна кнопка «расчет». Расчет с клиентом за комплексный обед производится в кредит, при этом специальная карточка с нанесенным на нее штриховым кодом клиента вставляется в считывающее устройство, где после нажатия кнопки «расчет» происходит считывание телевизионного изображения кредитной карточки в память ЭВМ, распознавание и сравнение с имеющейся информацией о данном кредитном номере. Если предъявленный кредитный номер разрешен к расчету в кредит, на турникет, открывающий доступ клиенту в зону раздачи, подается сигнал открытия, а данные о выбранном комплексе в форме 80-битовой посылки (номер линии, вид расчета (кредит), кредитный номер, номера заказанных блюд, стоимость всего заказа, время дня на момент заказа) передаются в центральный комплекс для дальнейшего использования (в данной версии для ведения протоколов).

Если кредитный номер запрещенный, на терминал комплекса выводится сообщение «запрещенный кредитный номер» (информация в центральный комплекс не передается, сигнал на открытие турникета не вырабатывается). Периферийный комплекс переходит в исходное состояние. Время начальной загрузки для одного периферийного комплекса — 30...40 с, время передачи информации о заказе из периферийного комплекса в центральный — 3 с.

Как одна из вспомогательных разработана программа для линии свободного выбора с расчетом в кредит. На этой линии клиент столовой может выбрать в единичном количестве любое из представленных на терминале блюд, манипулируя двумя кнопками на пульте пользователя: перемещения маркера и выбора/отмены. По окончании выбора клиент нажимает кнопку «расчет», и далее система работает аналогично линии комплексных обедов.

Для окончательного расчета с клиентами столовой, с целью удержания из заработной платы стоимости обедов в кредит, используется файл протоколов, сформированный на «Электронике НГМД-6022» центрального комплекса. На устройстве, описанном в [2], происходит перезапись файла протокола с гибкого диска на магнитную ленту и перекодировка информации в формат ЕС ЭВМ. При расчете заработной платы на ЕС ЭВМ для каждого работника предприятия учитывается сумма, затраченная на питание в столовой. Таким образом, САКР состоит из двух уровней: комплекса аппаратно-программных средств САКР и АСУ предприятия. Программное обеспечение реализовано на языке Паскаль. Общий объем 5000 операторов.

Телефон 532-86-90, Москва

ЛИТЕРАТУРА

1. Комплектный класс технических средств на базе микроЭВМ «Электроника БК-0010Ш» и ДВК-2МШ / Г. И. Фролов, С. М. Косенков и др. // Микропроцессорные средства и системы.— 1986.— № 4.— С. 65—66.
2. Чернявский А. Д., Шаповалов В. И. Комплексирование микроЭВМ «Электроника НЦ-8020» и НМЛ. // Микропроцессорные средства и системы.— 1987.— № 2.— С. 56.

3. Тенденции развития диалоговых вычислительных комплексов / В. С. Копорин, Л. С. Кридинер и др. // Микропроцессорные средства и системы.— 1986.— № 4.— С. 11—15.

Статья поступила 8.04.88

УДК 681.1.327

А. В. Савостьянов, Ю. Н. Давыденко, А. Н. Кузнецов

ПРОСТОЙ ПУЛЬТОВЫЙ ДИСПЛЕЙ

В системах с небольшим объемом выводимой информации и повышенными требованиями к механической прочности, надежности и габаритам успешно применяется БИС контроллера клавиатуры и дисплея КР580ВВ79, позволяющая полностью освободить МП от операций сканирования клавиатуры и регенерации отображения на дисплее. Пультдовый дисплей на основе микросхемы КР580ВВ79 и светодиодных матриц АЛС340А полностью совместим с шиной MULTIBUS и позволяет отображать цифровую и символьную информацию (см. рисунок).

Технические характеристики пультового дисплея

Число символов в строке	16
Число строк	1
Напряжение источника питания, В	5±5%
Потребляемый ток (макс.), А	1
Число клавиш	64
Код	КОИ8
Возможность работы по прерыванию	Есть
Емкость ОЗУ, Кбайт	8
Режимы работы	Бегущая строка и обычный

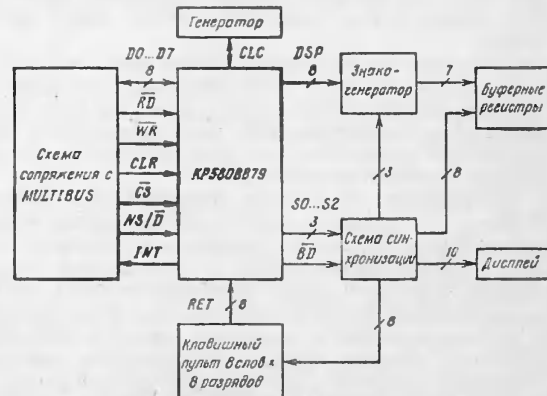
Микросхема КР580ВВ79 обеспечивает выдачу кода знака, сигналов сканирования клавиатуры и дисплея, связи с системой. Знакогенератор формирует необходимое изображение в соответствии с кодом знака, поступающим с БИС КР580ВВ79 в память (К573РФ2).

«Прошивка» знакогенератора соответствует расширенному КОИ8, включающему греческие буквы, а также некоторые другие символы. При необходимости знакогенератор может быть перепрограммирован под любой другой код обмена информацией.

Схема синхронизации, выполненная на микросхемах серий К555 и К561, обеспечивает синхронную работу всех составных частей дисплея, объединяющего 16 35-элементных светодиодных матриц. Клавишный пульт насчитывает 64 кнопки ПКн-125.

Буферные регистры собраны на восьми микросхемах КР580ИР83. Если вместо регистров использовать К555ИР22 или К555ИР23, то можно уменьшить ток, потребляемый дисплеем. Для этого необходимо перепрограммировать знакогенератор, проинвертировав его содержание. Схема сопряжения с шиной MULTIBUS реализована на микросхемах серий КР580 и К555.

Режимы инициализации КР580ВВ79, вывода символа на



Электрическая структурная схема дисплея

дисплей, ввода символа без эха, ввода символа с эхом, вывода сообщения и другие поддерживаются программной-драйвером.

Конструктивно дисплей вместе с индикаторами и клавиатурой выполнен в виде выносного пульта размерами 220×80×40 мм. Клавиатура представляет собой матрицу 4×16 линий, в узлах которой расположены кнопки, работающие на замыкание. Защитная пленка с нанесенными на нее символами предохраняет клавиатуру от повреждений. Перекодировка клавиш осуществляется программно, что позволяет применять дисплей в специализированных устройствах.

Телефон 535-26-71, Москва

ЛИТЕРАТУРА

1. USA, INTEL Corp., 1977, 11 p. 8279 Programmable Keabord/Display Interface.
2. Кулешова В. И. Микропроцессорный комплект БИС серии КР580. // Микропроцессорные средства и системы. — 1988. — № 3. — С. 88—93.

Сообщение поступило 10.10.88

УДК 681.326

С. И. Черняк, В. М. Табаткин

ИНДИКАТОР ШИН МИКРОПРОЦЕССОРА К1810ВМ86

Индикатор шин (ИШ) предназначен для аппаратной настройки устройств на основе МП К1810ВМ86 и К1810ВМ88 [1]. Индикатор состоит из микропульта, платы процессора и сорокаштырьковой вилки (рис. 1). Вилка

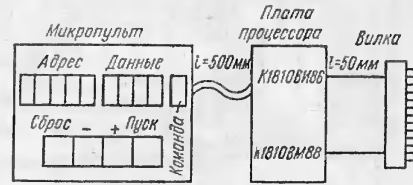


Рис. 1. Общая структура индикатора шин

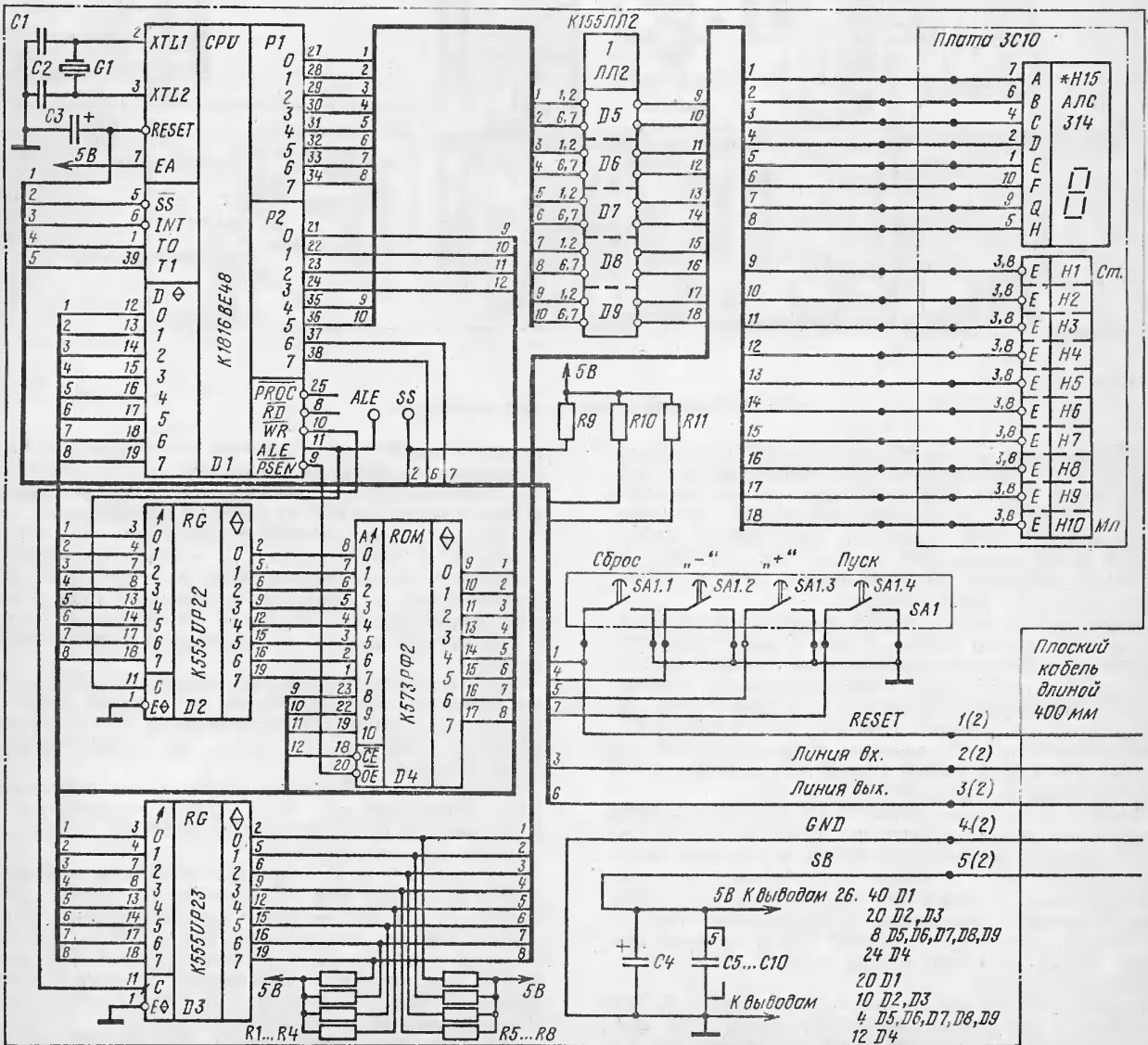


Рис. 2. Принципиальная схема микропульта

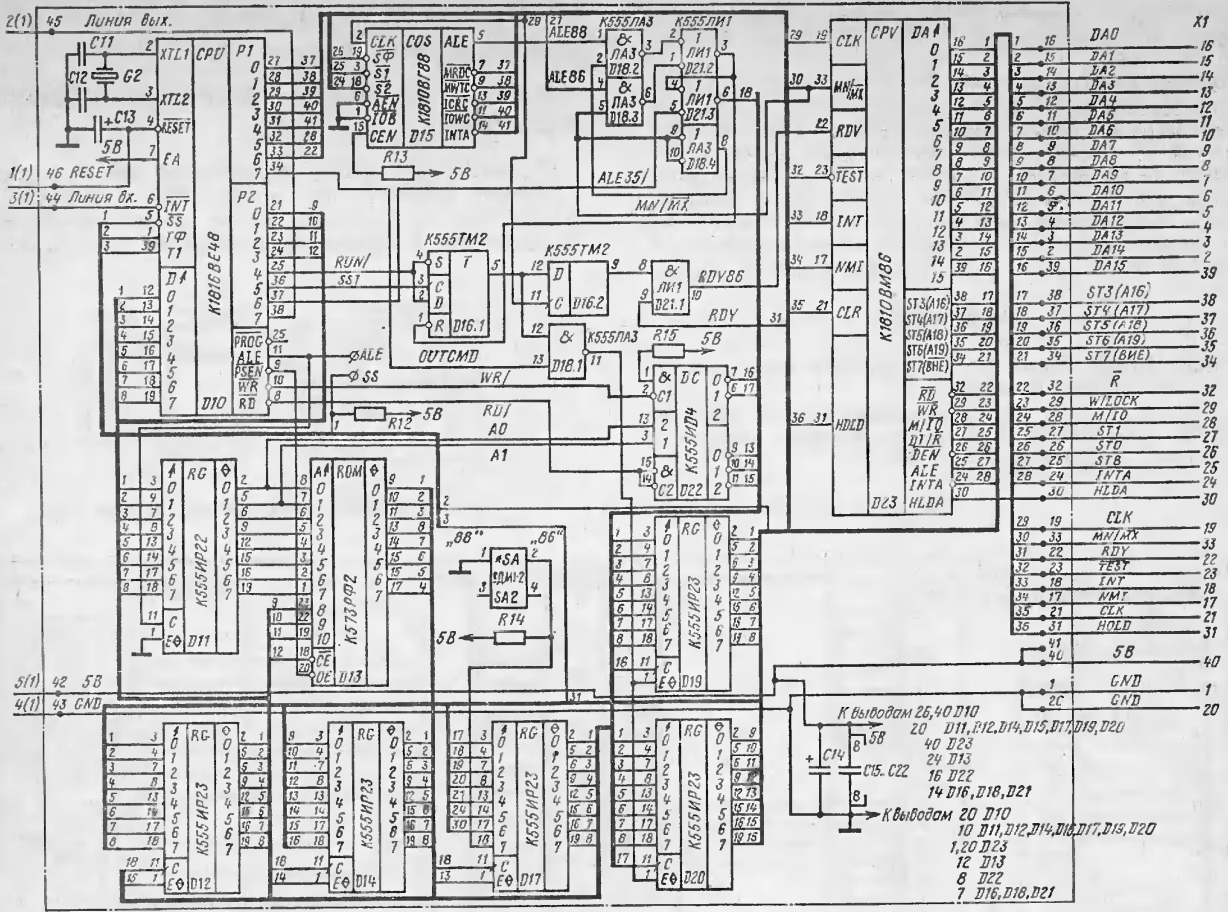


Рис. 3. Принципиальная схема процессора

устанавливается в розетку МП настраиваемого устройства. Питание ИШ осуществляется через соответствующие контакты вилки. Плата процессора содержит эмулирующий МП К1810ВМ86 или К1810ВМ88, который всеми выводами, за исключением RDY, подключен через плоский кабель длиной 50 мм к вилке ИШ. Сигнал RDY, поступающий с платы пользователя, «перехватывается» ИШ для обеспечения пошагового режима работы.

Микроузел соединен с платой процессора пятипроводным кабелем длиной 500 мм. С него в плату процессора поступают в последовательном коде соответствующие команды, а с платы также в последовательном коде выдается в микроузел информация о состоянии шины адреса, данных и управляющих сигналах эмулирующего МП.

Микроузел имеет 10-разрядный цифровой дисплей для отображения адреса, данных и управляющих сигналов МП и четыре клавиши управления работой ИШ.

Клавиша «Сброс» предназначена для физического сброса микроузла и платы процессора. Клавиши «←» и «→» необходимы для перемещения запятой по разрядам цифрового дисплея влево и вправо соответственно, а также для увеличения и уменьшения шестнадцатеричной цифры соответствующего индикатора. Кратковременное нажатие на клавишу «←» или «→» (менее 1/3 с) изменяет значение шестнадцатеричной цифры, а более длительное — перемещает запятую.

Клавиша «Пуск» служит для запуска индикатора шин в работу в выбранном режиме. По сбросу ИШ переходит в исходное состояние и на его правом крайнем индикаторе появляется символ выбранного режима. По

нажатию клавиши «←» или «→» последовательно перебираются все возможные режимы работы ИШ. После выбора соответствующего режима можно нажать клавишу «Пуск» и перевести ИШ на работу в этом режиме.

Среди возможных режимов работы ИШ — Т — пошаговый, G — автоматический, q — полуавтоматический с остановом на контрольной точке, P — установки контрольной точки, A — перехода по заданному адресу, H — полуавтоматический с остановом на контрольной точке и подстановкой данных, U — пошаговый с подстановкой данных.

Опрос клавиатуры, управление индикацией и связь с платой процессора по двухпроводной линии осуществляет однокристалльная ЭВМ (ОЭВМ) К1816ВЕ35 под управлением монитора микроузла (рис. 2), «защитого» в РПЗУ К573РФ2.

Выдачу всех необходимых управляющих сигналов на плату процессора (рис. 3) и связь с микроузелом по двухпроводной линии обеспечивает ОЭВМ под управлением монитора процессорной платы.

Регистры D12 и D14 предназначены для сохранения адреса по сигналу ALE МП и данных по сигналу ALE35 ОЭВМ, регистр D17 — для сбора информации с управляющих выводов МП, регистры D19 и D20 — для подстановки кода команды и данных, системный контроллер D15 — для выработки управляющих сигналов в максимальном режиме. На триггере D16 собрана схема управления готовностью в пошаговом режиме.

При установке эмулирующего МП К1810ВМ86 или К1810ВМ88 выключатель SA2 необходимо перевести в со-

ответствующее положение. ИШ удобно применять при настройке и ремонте устройств на базе МП К1810ВМ86, прогона «защиту» в РПЗУ программу в автоматическом, пошаговом или полуавтоматическом режиме с остановом на контрольной точке.

Если неисправность заключается в «залипании» отдельных сигналов адреса, данных или управляющих сигналов, то ее удается локализовать за первые же несколько шагов в пошаговом режиме. При начальной настройке устройства, когда нет готовых программ в ПЗУ, с помощью подстановки данных можно запрограммировать предусмотренные на плате ВУ и проверить ОЗУ.

Режим подстановки данных позволяет использовать ИШ как оценочное устройство при начальном знакомстве с работой МП К1810ВМ86 [2].

Телефон 22-28 05, Омск

ЛИТЕРАТУРА

1. Кобылинский А. В., Москалевский А. И., Темченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86 // Микропроцессорные средства и системы.— 1986.— № 1.— С. 28—33.
2. Микропроцессоры: Средства отладки, лабораторный практикум и задачник: Учеб. для вузов. Н. В. Воробьев, В. Л. Горбунов, А. В. Горячев и др.; / Под ред. Л. Н. Преснухина.— М.: Высшая школа, 1986.— Кн. 3.— С. 81.

Статья поступила 1.09.88

ЛОКАЛЬНЫЕ СЕТИ И СРЕДСТВА МЕЖМАШИНОЙ СВЯЗИ

УДК 681.324.01

А. М. Дьяченко, В. М. Королев, С. И. Симаненков, Б. В. Цейтлин

СТАНЦИЯ ЛОКАЛЬНОЙ СЕТИ «ЭЛЕКТРОНИКА МС8301»

При создании гибких производственных модулей возникает задача информационного объединения различного оборудования: устройств числового программного управления, систем управления автоматизированными складами, диагностических систем, станций подготовки программ и др. Для решения указанной задачи повсеместное применение находят локальные вычислительные сети (ЛВС). При построении таких сетей технологическо-

го оборудования могут быть использованы станции локальной сети «Электроника МС8301» (СЛС). С помощью одной станции можно подключить к кольцевой сети до восьми абонентских устройств (рис. 1).

Рассмотрим реализацию уровней в СЛС «Электроника МС8301» [1]. В основании иерархической пирамиды располагается среда передачи данных — серийно выпускаемое устройство цифровой передачи данных «Элект-

диагностики, остальные — интервалы восстановления маркера, маркерный интервал управления и диагностики для передачи команд администратора (администратором сети назначается одна из станций).

На уровне логического канала реализуется обмен с помощью кадров, структура которых приведена на рис. 2. Заголовков и концевик кадра разбиваются на поля длиной два байта (поле данных — поле переменной длины с четным числом байтов от 2 до 4096), на канальном уровне не анализируются и не изменяются.

Передаваемые кадры встречаются двух типов: кадры записи принимаются станцией адресатом и переносят данные более высокого (транспортного) уровня; логические кадры не принимаются, но в них модифицируются поля данных проходящего кадра («на лету») — выполняется поразрядная операция И над полем данных кадра и данными, находящимися в специальном запоминающем устройстве станции-адресата. Это свойство используется для синхронизации станций сети при решении задачи, распределенной между несколькими узлами. Следует отметить, что обмен логическими кадрами — нетрадиционная для сетей ЛВС, расширяющая возможности ЛВС.

Адрес приемника — адрес станции-адресата. Каждая станция сети может обладать индивидуальными и несколькими групповыми адресами. Два младших разряда указывают тип адреса, остальные четырнадцать задают его значение.

Счетчик узлов указывает длину поля данных в байтах и число станций локальной сети, принявших данный кадр (значение поля может модифицироваться станцией-адресатом).

Признаки приема модифицируются станциями-адресатами кадра, которые выполняют операцию И над полем признаков приема кадра и внутренним регистром признаков приема, содержат биты, подтверждающие прием кадра либо указывающие причину отсутствия.

Характеристики СЛС «Электроника МС8301» и сети на ее основе

Среда передачи	Опволоконный кабель
Скорость передачи, Мбит/с	1,6
Вид топологии сети	Однонаправленное кольцо
Расстояние между станциями, м, не более	300
Число станций локальной сети не более	512
Метод доступа	Передача маркера
Метод защиты от ошибок	Циклический контрольный код на основе образующего полинома $X^{16} + X^{12} + X^6 + 1$
Интерфейс с АУ	Токовая петля либо стык С2
Скорость обмена с АУ по каналу, бит/с, не более	19 200
Реализуемые сетевые уровни в соответствии с моделью открытых систем ISO/TC 97/C	Физический, канальный транспортный, сеансовый
Габаритные размеры станции, мм	350×448×233

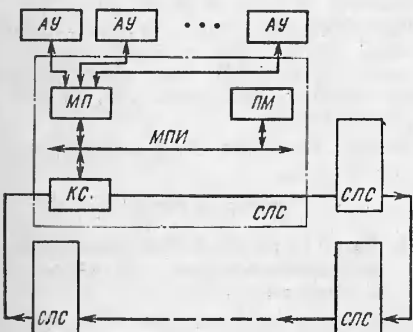


Рис. 1. Конфигурация локальной сети на базе СЛС «Электроника МС 8301». АУ — абонентное устройство, КС — контроллер сети, ПМ — процессорный модуль, МП — мультипроцессор

роника МС4101» (ЦСПД), состоящее из приемника-преобразователя, передатчика-преобразователя и волоконно-оптической линии связи (ВОЛС).

На следующем физическом уровне данные передаются по линии связи по принципу временного разделения: каждый вид информации за определенный промежуток времени, называемый канальным интервалом (за один канальный интервал сообщается один байт данных). Время трансляции данных всех 16 интервалов — канальный цикл. Десять канальных интервалов занимают интервалы данных, два — тестовые интервалы непрерывной аппаратной

Заголовок кадра					Концевик кадра		
Тип кадра	Адрес приемника	Счетчик узлов	Адрес источника	Счетчик байтов	Данные	Признаки приема	Контрольная последовательность

Рис. 2. Структура кадра

Все поля кадра, предшествующие полю «Контрольная последовательность», проверяются циклическим 16-разрядным избыточным кодом.

Контроллер сети (КС) выполнен на двух платах с использованием микросхем серий К555, К531, К556, К559, К1802 и К1002. Его основные функции — прием, передача и обработка кадров. В каждом проходящем по сети кадре КС сравнивает поле адреса назначения с набором собственных адресов, хранящихся в специальном запоминающем устройстве (32 байт): при наличии адресованного кадра он записывается в режиме ПД в один из буферов, организованных в ОЗУ процессорного модуля.

Для предотвращения потери принимаемых кадров (при асинхронном поступлении на станцию) служит запоминающее устройство типа FIFO, хранящее очередь адресов свободных буферов. При поступлении кадра адрес буфера автоматически выбирается из этой очереди (вытесняется из FIFO) для программирования операции ПДП. Очередь адресов пополняется программной записью в FIFO, одновременно с ОЗУ подсчитывается контрольная последовательность, которая сравнивается со значением концевика кадра. При несовпадении величин кадр, записанный в ОЗУ, не принимается (не происходит вытеснения адреса из очереди); буфер вновь считается свободным.

Данные из ОЗУ процессорного модуля считываются в режиме ПДП, оформляются в виде кадров и передаются в сеть (КС может передавать и принимать кадры одновременно).

Важная функция КС — поддержка функционирования маркера сети. Маркер представляет собой код, передаваемый в маркерном интервале. Общепринято сравнивать прохождение маркера по сети с передачей эстафетной палочки. Обладание такой «палочкой» дает станции право на передачу кадров. В любой момент времени не может быть более одной передающей СЛС. Таким образом, маркер — средство, регулирующее доступ станции к сети [2]. В аварийной ситуации начинается процедура восстановления маркера, в которой участвуют все станции.

Процессорный модуль (ПМ) используется для программной реализации сетевых протоколов верхних уровней и управления модулями КС и МП. ПМ станции локальной сети — одноплатная микроЭВМ с объемом ОЗУ — 256 Кбайт, ПЗУ — 128 Кбайт, быстро-

действием — 600 тыс. оп./с типа регистр-регистр и двумя последовательными портами для обмена данными с периферийным оборудованием по токовой петле. МикроЭВМ имеет выход на стандартную магистраль МПИ с 22-разрядным адресом.

В качестве процессора ПМ использован МП К1801ВМ3, обладающий диспетчером памяти и системой команд, аналогичной мини-ЭВМ «Электроника 100/25». ОЗУ ПМ выполнено на БИС К565РУ5 с устройством управления на УВМ К1801ВП1-30. ПЗУ предусмотрено двух видов: системное с программами, обеспечивающими старт микропроцессора, работу в режиме пульта терминала, загрузку через последовательный порт и хранения программ.

Распределение адресного пространства ПМ

ОЗУ	0...777 776
ПЗУ хранения программ	4000000...4377776
Системное ПЗУ	17570000...17 577776
Регистры внешних устройств	c 177600000

Процессорный модуль выполнен на одной плате с использованием микросхем серий К1801, К565, К556, К558, К559, К531, К555.

Восьмиканальный асинхронный мультиплексор (МП) обеспечивает обмен в 5-, 6-, 7-, 8-битовом старт-стопном коде с возможностью формирования бита, дополняющего до четности или нечетности, и выбора числа столбцовых элементов. Конкретный формат передаваемых данных по каждой линии устанавливается отдельно с помощью движковых переключателей или программно записью в регистр. Скорость обмена для каждой линии выбирается из ряда: 150, 300, 600, 1200, 2400, 4800, 9600, 19200 бит/с.

Данные, принимаемые по восьми каналам, а также номера каналов, по которым эти данные были приняты, поступают в запоминающее устройство типа FIFO. Накопление данных сглаживает пиковые нагрузки на процессор при одновременном поступлении данных от нескольких абонентских устройств (АУ). Линейная часть МП осуществляет обмен по токовой петле и стандарту стьк С2 (выбор конкретного стандарта производится для каждого канала отдельно с помощью движкового переключателя).

Структура регистров МП в основном аналогична DZ 11 фирмы DEC (США), за исключением регистров управления модемом.

Мультиплексор выполнен на одной плате с использованием микросхем се-

рий К1002, К581, К559, К531, К555, 5170.

Средства контроля и диагностики станции можно разделить на две группы: средства, контролирующие работу станции непосредственно во время эксплуатации, и программы экспресс-диагностики после включения питания СЛС, определяющие неисправность во время ремонта и проведения профилактических работ.

К средствам первой группы можно отнести:

схемы циклического контроля, аппаратно реализованные в блоке контроллера;

специальный интервал данных, по которому непрерывно передается тестовая информация, анализируемая каждой станцией сети (строго установленная всем станциям последовательность битов).

После включения станции начинают выполняться в сокращенном варианте резидентные тесты в ПЗУ ПМ, которые последовательно проверяют исправность всех модулей СЛС, правильность выполнения команд микропроцессором, функционирование ОЗУ, совпадение контрольной суммы ПЗУ с

эталонной, передачу кадров по внутреннему шлейфу. Замыкание канала информации передачи данных из станции в сеть внутри самой станции позволяет проверить весь тракт обработки данных; кольцо локальной сети при этом не разрывается, так как данные сети ретранслируются через станцию без какой-либо обработки их в станции. Проверка МП заключается в передаче данных по внутреннему шлейфу, замыкающему вход и выход каждого канала.

Для полного контроля используются подробные резидентные тесты. Подключение диагностического дисплея к одному из портов ПМ запускает тесты в режиме диалога с оператором.

В настоящее время готовится к серийному выпуску еще одно исполнение станции, в которой будет установлен интерфейс для связи с АУ, имеющими шину Unibus. Это позволит подключить к сети ЭВМ типа «Электроника 100/25», «Электроника 79», «Электроника 82».

214015, Смоленск, ОКБ «Меркурий», тел. 1-11-25

ЛИТЕРАТУРА

1. Якубайтис Э. А. Информационно-вычислительные сети. — М.: Финансы и статистика, 1984.
2. Флинт Д. Локальные сети ЭВМ: архитектура, принципы построения, реализация. — М.: Финансы и статистика, 1986.

Статья поступила 4.04.88

Ю. В. Новиков

УНИВЕРСАЛЬНЫЙ ПАРАЛЛЕЛЬНЫЙ ИНТЕРФЕЙС ДЛЯ МОДУЛЬНЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ ИЗМЕРЕНИЯ, КОНТРОЛЯ И УПРАВЛЕНИЯ

В широко применяемых устройствах параллельного обмена, в частности И2, невысокая разрядность внешних шин (16 входных и 16 выходных линий) и жесткое задание направления передачи по этим шинам снижают эффективность использования. Для увеличения гибкости системы при снижении аппаратных затрат необходимо применение интерфейса с большим числом внешних линий и простотой изменения направления передачи по этим линиям (организуется двунаправленный обмен данными).

Разработанный универсальный параллельный интерфейс (см. рисунок) обеспечивает обмен информацией в программном режиме по 64 двунаправленным линиям, образующим четыре 16-разрядных порта. Каждому порту соответствует свой адрес в адресном пространстве микроЭВМ. Интерфейс позволяет реализовать режимы ввода, вывода информации, а также вывода с проверкой правильности передачи информации по каждому из четырех портов. Предусмотрено программное переключение направления передачи внешних линий с точностью до одного байта (одна из половин порта может использоваться для ввода информации, другая — для вывода).

Внешние линии интерфейса — двунаправленные, поэтому на линию в режиме вывода информации может поступить внешний сигнал с логическим уровнем, не совпадающим с уровнем соответствующего выходного сигнала интерфейса (например, интерфейс выдает на линию сигнал, а внешний сигнал имеет уровень Лог. 0). При этом возможен выход из строя как выходного каскада интерфейса, так и источника внешнего сигнала. Эта ситуация характерна для систем контроля и отладка внешних устройств. Для предотвращения выхода из строя интерфейса предусмотрена защита внешних линий:

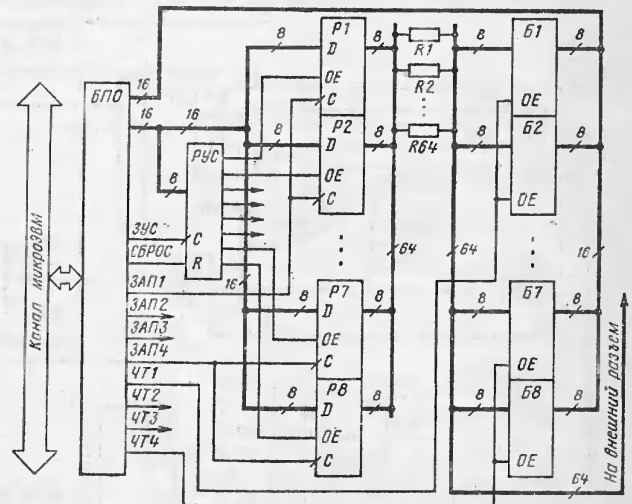
при включении питания все 64 внешние линии переключаются в режим приема (по сигналу СБРОС, вырабатываемому БПО, сбрасывается РУС);

при выводе информации из интерфейса правильность передачи проверяется программно с помощью ее чтения из соответствующего порта. При несовпадении передаваемой и принимаемой информации выходные линии программно отключаются (переводятся в режим приема информации извне);

для ограничения уровня выходного тока сигналы с выходов регистров P1...P8 на внешний разъем подаются через резисторы R1...R64 с номиналами 30...100 Ом. Пользователь может определить причину прихода внешних сигналов противоположного уровня без риска выхода из строя как интерфейса, так и источника внешнего сигнала. Опыт эксплуатации интерфейса показал, что снижением нагрузочной способности выходных регистров и помехоустойчивости в большинстве случаев можно пренебречь.

Входные буферы имеют выходы с тремя состояниями, что позволяет без дополнительных аппаратных затрат реализовать мультиплексирование принимаемых данных со всех четырех портов (выбор одного из портов производится сигналами ЧТ1...ЧТ4). Чтение информации из порта применяется для приема внешних сигналов и контроля правильности выдачи информации интерфейсом по этому же порту.

Внешние линии интерфейса переключаются программно на прием информации переводом выходных каскадов регистров P1...P8, имеющих выходы с тремя состояниями, в высокоимпедансное состояние по входам ОЕ. Управляющее слово, определяющее направление передачи каждого из восьми байт 64-разрядного слова, программно записывается в РУС по сигналу ЗУС, вырабатываемому БПО. При этом каждый из восьми битов управляющего слова задает направление передачи одного из восьми байтов. С помощью программного управления направлением передачи может быть реализован режим одновременной выдачи большого числа сигналов (до 64) при записи в выходные регист-



Структурная схема универсального параллельного интерфейса:

БПО — блок программного обмена, РУС — регистр управляющего слова (2xK155TM8), P1...P8 — выходные регистры (K555IP23), B1...B8 — входные буферы (K555IP22), ЗУС — запись управляющего слова, ЗАП1...ЗАП4 — запись информации в порты 1...4, ЧТ1...ЧТ4 — чтение информации из портов 1...4, СБРОС — сигнал сброса РУС

ры с отключенными выходами требуемых сигналов и последующем переводе выходов регистров в активное состояние.

На внешний 90-контактный разъем интерфейса выведены 64 информационные линии и шины питания, а также управляющие сигналы ЗУС, СБРОС, ЗАП1...ЗАП4, ЧТ1...ЧТ4. Эти сигналы информируют устройство, с которым осуществляется сопряжение, о передаче информации с интерфейса или о ее приеме, что в ряде случаев упрощает сопряжение.

Универсальный параллельный интерфейс реализован на полуплате микроЭВМ «Электроника 60» и содержит 30 корпусов широко распространенных интегральных микросхем малой и средней степени интеграции.

115409, Москва, Каширское шоссе, 31, МИФИ;
тел. 324-95-65

Статья поступила 24.03.88

УДК 681.3(075.8)

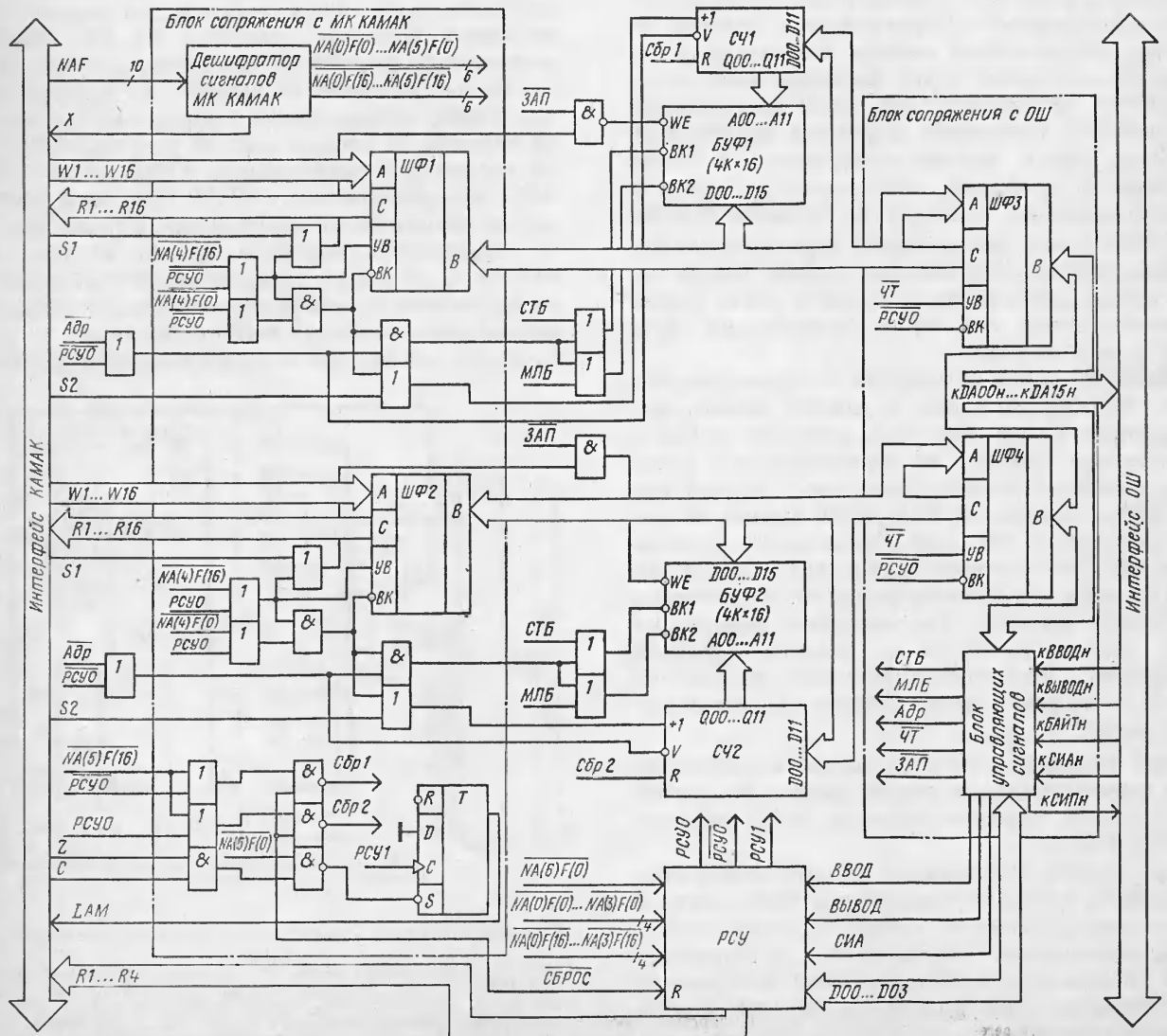
М. А. Каявченко, В. В. Малахов, А. С. Федулов,
В. В. Борисов

УСТРОЙСТВО ОБМЕНА МАССИВАМИ ДАННЫХ МЕЖДУ ИНТЕРФЕЙСАМИ КАМАК И ОШ

Устройство используется в системе обработки данных на основе мини-ЭВМ, имеющей канал типа ОШ, реализованный в стандарте КАМАК. Оно обеспечивает двусторонний обмен массивами 16-разрядных данных объемом до 4 Кслов. Управление обменом возможно как со стороны контроллера крейта КАМАК, так и от мини-ЭВМ. Система обработки может содержать несколько устройств обмена, в зависимости от тела свободных мест в крейте, к каждому из которых подключена своя мини-ЭВМ.

В состав устройства (см. рисунок) входят два буферных накопителя БУФ1 и БУФ2, счетчики адресов СЧ1 и СЧ2, блок сопряжения с МК КАМАК, блок

Функциональная схема устройства обмена



Назначение разрядов РСУ

Регистр	Разряд	Назначение
PCU 0	1	Подключение БУФ1 к МК КАМАК, БУФ2 к ОШ
	0	Подключение БУФ2 к МК КАМАК, БУФ1 к ОШ
PCU1	1	Буфер, подключенный к ОШ, заполнен
	0	Буфер, подключенный к ОШ, не заполнен
PCU2	1	Буфер, подключенный к МК КАМАК, заполнен
	0	Буфер, подключенный к МК КАМАК, не заполнен
PCU 3	1	Ошибка в устройстве
	0	Устройство исправно

сопряжения с интерфейсом ОШ, регистр состояния устройства РСУ.

Обращение к ячейкам памяти со стороны МК КАМАК осуществляется последовательно, начиная с нулевого адреса. Со стороны ОШ буферная память имеет произвольный доступ и занимает зону адресов, соответствующую одному из банков памяти ОЗУ мипи-ЭВМ, и заменяет его. Предусмотрена возможность побайтной адресации. В любой момент времени один из буферных накопителей подключен к каналу ОШ, а второй — к МК КАМАК. Обмен данными происходит путем их переключения. Направление включения каждого накопителя определяется РСУ.

УДК 681.3

А. И. Князев, С. М. Шестимеров

ЭЛЕКТРОННЫЙ ДИСК С ПРЯМЫМ ДОСТУПОМ К ПАМЯТИ ДЛЯ ДВК2М

В качестве электронного диска для ДВК2М использовано полупроводниковое внешнее запоминающее устройство СМ5902, предназначенное для хранения больших массивов данных и организации обмена при работе в составе комплекса СМ ЭВМ. Его можно считать электронным аналогом дисковой системы с фиксированными головками. Выпускается серийно с 1986 г.

Основные технические характеристики устройства

Интерфейс	ОШ ОСТ 25 795-78
Режим работы интерфейса	ПДП
Режим работы	Чтение, запись, контроль записи
Емкость, Мбайт	4
Разрядность слова, бит	16 плюс 6 контрольных
Средняя скорость передачи, Мбит/с, не менее	13,3
Среднее время доступа, мкс/слово, не более	1
Цикл обращения к регистру, мкс, не более	0,75
Периодичность режима регенерации, мкс, не более	7,5

В СМ5902 обеспечен режим самодиагностики с обнаружением двойной ошибки, коррекцией одиночной и двойной ошибок типа сбоя на фоне устойчивого отказа или двух устойчивых отказов.

Связь ДВК2М, имеющего магистраль МПИ, с СМ5902 организована с помощью разработанного для этой цели интерфейса (рис. 1). Регенерация ОЗУ ДВК2М реализована без дополнительных аппаратных затрат, так как обмен между ДВК2М и СМ5902 в режиме ПДП осуществляется словами. Отпала необходимость в применении диспетчера памяти, поскольку два старших разряда адреса регистров ОША16 и ОША17 постоянно выбраны. Появилась возможность отказаться от использования таких сигналов интерфейса ОШ, как ОШК1, ОШК2 и ОШ ПРЕР.

Интерфейс состоит из шести функциональных блоков. Регистр адреса

Блок сопряжения с МК дешифрует команды КАМАК (см. ниже), реализует обмен данными между магистралью и буферными накопителями, осуществляет чтение РСУ в МК КАМАК.

Дешифрация команд КАМАК

Команда	Назначение
NA (0)F(0)	Установка РСУ 0 в 1
NA (1)F(0)	Установка РСУ 1 в 1
NA (2)F(0)	Установка РСУ 2 в 1
NA (3)F(0)	Установка РСУ 3 в 1
NA (4)F(0)	Чтение 16-разрядного слова из буфера, подключенного к МК КАМАК, на шины R1...R16
NA (5)F(0)	Сброс LAM устройства обмена и чтение РСУ на МК КАМАК
NA (0)F(16)	Сброс РСУ 0 в 0
NA (1)F(16)	Сброс РСУ 1 в 0
NA (2)F(16)	Сброс РСУ 2 в 0
NA (3)F(16)	Сброс РСУ 3 в 0
NA (4)F(16)	Запись 16-разрядного слова в буфер, подключенный к МК КАМАК с шин W1...W16
NA (5)F(16)	Сброс счетчика адреса буферного накопителя, подключенного к МК КАМАК.

Четырехразрядный РСУ задает режимы работы буферных накопителей, определяет их готовность к обмену (см. таблицу). Адрес РСУ со стороны ОШ находится в зоне адресов внешних устройств канала ОШ.

Блок сопряжения с интерфейсом ОШ обеспечивает обмен данными между буферными накопителями и ОШ, вырабатывает управляющие сигналы устройства обмена в соответствии с управляющими сигналами ОШ, реализует доступ к РСУ со стороны ОШ.

Устройство можно использовать в качестве буферного устройства обмена массивами данных между интерфейсами МК КАМАК и ОШ и в качестве буферного ОЗУ МК КАМАК.

Устройство обмена реализовано на ИМС средней степени интеграции серий К155, К541, К589. Конструктивное исполнение — одиночный блок КАМАК.

214013, Смоленск, Энергетический проезд, 1, Смоленский филиал МЭИ; тел. 9-11-90

Статья поступила 27.06.88

(РА) предназначен для запоминания адреса, выставляемого на шину интерфейса ДВК2М, и для разделения адресной шины (ОША) и шины данных (ОШД). Записанный в РА адрес дешифрируется с помощью дешифратора адресов (ДА), который служит для селекции адресов рабочих регистров СМ5902. Разделение шин и электрическое согласование обеспечиваются бу-

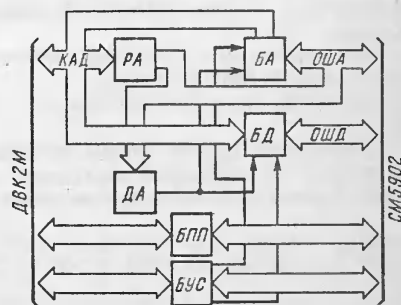


Рис. 1. Структурная схема интерфейса

ферными элементами (БА — буфер адреса, БД — буфер данных).

Блок преобразования сигналов управления и синхронизации (БУС) предназначен для дешифрации сигналов управления и синхронизации а также формирования временной диаграммы в соответствии с протоколом обмена между МПИ и ОШ, блок прерывания и прямого доступа к памяти (БПП) — для формирования сигналов, обеспечивающих режимы прерывания и ПДП. Интерфейс выполнен на сорока микросхемах средней и малой степени интеграции, размещенных на одной плате микроЭВМ «Электроника 60».

Функционирование СМ5902 в ОС ДВК поддерживается с помощью монитора RFMNSJ и программы драйвера с именем RF. Исходный текст драйвера (рис. 2) транслируется, как обычная программа. Формируются две загрузочные дискеты для НГМД6022,

на которых организуется необходимый для работы и запуска системы набор программ. На каждой дискете должны быть стартовые файлы типа СОМ (рис. 3, а, б).

Работать с СМ5902 надо, как с обычным системным устройством, а НГМД служит только для хранения рабочих программ. При необходимости пользователь может дополнить набор программ и данных на СМ5902 копированием с НГМД.

Комплекс на базе ДВК2М и СМ5902 с интерфейсом позволяет значительно

разгрузить НГМД, что в десятки раз увеличивает срок службы магнитных головок и дискет; эффективно работать в системе с языками Паскаль, Фортран 4, Си; повысить производительность и удобство работы программиста, так как уменьшается трудоемкость разработки программ (редактирование, компиляция, отладка, копирование).

Для сравнения ниже приведены характеристики работы комплексов ДВК2М — СМ5902 и ДВК2М — НГМД6022.

	ДВК2М — НГМД6022	ДВК2М — СМ5902
Время трансляции файла программы объемом 12 блоков, написанной на языке Си (компиляция, ассемблирование, линкование), мин	12,5	1,75
Время копирования файла объемом 432 блока, с	102	1

РЕКЛАМА

ПРЕДЛАГАЕМ ПРИНЦИПИАЛЬНЫЕ ЭЛЕКТРИЧЕСКИЕ СХЕМЫ 16-РАЗРЯДНОЙ ЭВМ В СТАНДАРТЕ VME

Центральный процессор:

микропроцессор K1801BM3 с сопроцессором плавающей арифметики K1801BM4; 128 Кбайт двухпортовой статической памяти; два последовательных порта на плате

Контроллер накопителей на магнитных дисках: двухпортовый буфер на дорожку диска (8 Кбайт);

до трех накопителей типа винчестер с интерфейсом ST506;

до двух накопителей на ГМД с интерфейсом SA450.

Фортран-процессор:

две VME-платы двойной высоты; микропрограммная реализация системы команд промежуточного кода компилятора Фортрана;

операционное устройство на микросхемах серии K1804 выполняет операции с 16- и 32-разрядными целыми числами с плавающей запятой; до 128 Кбайт ОЗУ программ и данных; около 1 млн. оп./с на программе Whetstone.

Контроллер последовательно-параллельных портов:

два параллельных (K580BB55) и шесть последовательных (K580BB51) портов; одна VME-плата двойной высоты.

Специалисты Центра готовы принять участие в наладке изготовленных заказчиком печатных плат, а также выполнить другие работы.

109240, Москва, 2-ой Котельнический пер., 3. НТТМ Дельта; тел. 272-35-61; 272-55-95; 272-56-17.

Рис. 2. Драйвер RF (начало)

```

; АССЕМБЛЕР MACRO 11/ОС ДВК
;*****
;* ДРАЙВЕР RF УСТРОЙСТВА СМ 5902 *
;*****

.MCALL .GELDF, .DRBEG, .DRFIN, .DREND, .DRAST, .FORK
.GELDF ;СИМВ. СМЕЩ. ДЛЯ ЭЛ-ТА ОЧЕРЕДИ
RF*VEC == 204 ;ВЕКТОР ПРЕРЫВАНИЯ
RFCS = 177460 ;РЕГИСТР СОСТОЯНИЯ
RWC = RFCS+2 ;СЧЕТЧИК СЛОВ
RFMA = RFCS+4 ;АДР. ШИНЫ
RFDA = RFCS+6 ;РАБ. АДРЕС ДИСКА
RFAE = RFCS+10 ;ТЕК. АДРЕС
CMAINH = 400 ;НАЧ. АДРЕС СПИСКА ПАРАМ.
RFDSIZ = 20000 ;РАЗМЕР УСТРОЙСТВА В БЛОКАХ
RFSTS = 100012 ;СТАТУС УСТРОЙСТВА (ПРЯМ.ДОСТУП)
HDERR = 1 ;БИТ АППАРАТНОЙ ОШИБКИ
RD = 105 ;РЕЖИМ ЧТЕНИЯ
WR = 103 ;РЕЖИМ ЗАПИСИ
RKSYS == 0 ;СТАНДАРТ. ПАРАМЕТРЫ ГЕНЕРАЦИИ
ILSYS == 0
ITSYS == 0
IXSYS == 0
DPSYS == 0
DSSYS == 0
DMSYS == 0
DYSYS == 0
RFIDEN = 5000
RFIDS = 5377

RFRCNT = 3400
RFNREG = 10
RFREGA = 177460

.DRBEG RF, RF*VEC, RFDSIZ, RFSTS ;ЗАГолоВок ДРАЙВЕРА
MOV #7, (PC)+ ;ИНИЦИАЦИЯ ВВОДА-ВЫВОДА
RFTRY: .WORD 0
AGAIN: JSR R0, RFCOMN
ZERO: .WORD 0
MOV (R5)+, -(R4)
MOV (R5)+, -(R4)
MOV #WR, R5
TST @R4
BEQ RFEXIT
BMI 1x
MOV #RD, R5
NEG @R4
1x: MOV R5, -(R4) ;ЗАПУСК УСТРОЙСТВА СМ 5902
RTS PC
.DRAST RF, 5, RFAE ;МЕТКА ПРОГРАММНОГО АННУЛИРОВ.
MOV RFQCE, R5 ;ТЕК. ЭЛЕМЕНТ ОЧЕРЕДИ
TST @#RFCS

BPL RFFILL
.FORK RFFBLK ;ОЧИСТКА СТЕКА ОТ ИНФОРМ. ПРЕП.
RFRETY: DEC RFRTRY
BGT AGAIN
RFRHER: BIS #HDERR, @-2(R5) ;УСТАНОВИТЬ БИТ АП-НОЙ ОШИБКИ
RFEXIT: .DRFIN RF

```



```

RFABORT: MOV    #400, @#RFCS      ;АННИЛИРОВАНИЕ ВВОДА/ВЫВОДА
          CLR    RFFBLK+2
          BR     RFXEXIT
RFFILL:   JSR    R0, RFCOMN      ;ЗАВЕРШЕНИЕ ВВОДА/ВЫВОДА
          .WORD  CMAINH
          MOV    2(R5), R5
          BPL   RFXEXIT
          TSTB  R5
          BEQ   RFXEXIT
          NEG   R5
          ADD   R5, @R4
          ADC   2(R4)
          BIS   #177400, R5
          MOV   PC, -(R4)
          ADD   #ZERO-, @R4
          MOV   R5, -(R4)
          MOV   #WR, -(R4)
          RTS   PC
POPHOM:  MOV    (SP)+, R0
          BR     RFXEXIT
RFCOMN:  MOV    #RFAE, R4      ;ОБРАБОТКА ЭЛЕМЕНТОВ ОЧЕРЕДИ
          BIT   @R0, @R4
          BNE  POPHOM
          MOV   #400, @#RFCS
          MOV   RFCQE, R5
          MOV   (R5)+, -(SP)
          SWAB @SP
          MOV   @SP, @R4
          BIC   #177740, @R4
          BIS   (R0)+, R4
          CLRB @SP
          MOV   (SP)+, -(R4)
          TST  (R5)+
          RTS   R0
RFFBLK:  .WORD  0, 0, 0, 0      ;ЭЛЕМЕНТ 0 ОЧЕРЕДИ
          .DREND RF
          .END

```

Рис. 2. Драйвер RF (окончание)

```

SET TT: QUIET
ASS DM1: DK
INST RF: /NOQ
INIT RF: /NOQ
COPY/SYS DM1: ** RF: **
COPY DM0: *.SAU RF: *.SAU
COPY DM0: *.H RF: *.H
COPY DM0: *.OBJ RF: *.OBJ
SET TT: SCOPE
COPY/BOOT RF: RFMNSJ.SYS RF:
BOOT RF:

```

Рис. 3. Стартовые файлы дискет № 1 (а) и № 2 (б)

Для повышения эффективности использования комплекса ДВК2М — СМ5902 целесообразно организовать многотерминальный режим работы системы. Для этого необходимо подключить к ДВК2М через дополнительную интерфейсную плату со своими системными адресами и вектором прерывания еще один терминал и включить в состав монитора RFMNSJ многотерминальный драйвер.

Телефон 468-81-75, Москва

Статья поступила 26.01.88

УДК 681.327

Н. И. Гайдашенко

РЕЗЕРВИРОВАННЫЙ ИНТЕРФЕЙС ДЛЯ УСТРОЙСТВ СВЯЗИ С ОБЪЕКТОМ

Управляющие вычислительные комплексы ПС1001 — следующий этап в развитии средств вычислительной техники линии АСВТ-М-СМ ЭВМ-АСВТ-ПС. Это оригинальная система аппаратных и программных средств без прямых зарубежных прототипов.

Основные отличительные особенности управляющих вычислительных комплексов ПС1001:

повышенная гибкость и широкие возможности наращивания;

улучшенные качественные характеристики — надежность, отказоустойчивость и живучесть комплекса за счет введения аппаратной и программной избыточности и глубокой диагностики и контроля.

В настоящее время средства для расширения возможностей и улучшения качества создаваемой системы приходится искать в области совершенствования архитектурных решений. При разработке ПС1001 вышеречисленные усовершенствования реализованы при использовании нового принципа организации связи с объектом. Его суть: блоки связи с объектом не выходят на системный интерфейс вместе с периферийным оборудованием, а образуют отдельный интерфейс низшего ранга — интерфейс резервированный (ИР).

Системный интерфейс комплекса и ИР соединяются с помощью блока управления связи с объектом БУСО и нескольких контроллеров ИР, которые устанавливаются в одном каркасе с модулями устройств связи с объектом (рис. 1).

Контроллер ИР может быть пассивным, выполняющим функции связи и разветвления сигналов, или активным с интеллектуальными возможностями. Протокол обмена с центральной частью одинаков, поэтому к одной магистрали

связи можно подключать как активные, так и пассивные контроллеры, обслуживая их одними и теми же системными программами.

ИР построен по магистрально-радиальному принципу и имеет следующие характеристики и возможности:

собственно ИР состоит из трех идентичных групп сигналов (направлений), которые функционируют по одинаковому алгоритму, но полностью развязаны друг от друга электрически и логически, поэтому ИР применяется в комплексах повышенной надежности (дублированных и троированных), так как при отказе одного направления не нарушается нормальная работа оставшихся;

все выходные сигналы каждого блока УСО на ИР определяются его входными сигналами (при снятии входных сигналов фиксируются выходные), что повышает устойчивость ИР к сбоям и уменьшает вероятность возникновения отказов на интерфейсных шинах;

информационные сигналы ИР передаются по 16 двуправленным магистральным шинам, управление — по однопроводным, сигналы контроллера «Выборка» и блока УСО «Запрос на прерывание» — по радиальным шинам (организация интерфейсной части блоков УСО упрощается и повышается отказоустойчивость ИР);

в одном конструктиве предусмотрено размещение до трех контроллеров и до 16 функциональных блоков УСО (один контроллер может обслуживать не более 16 блоков УСО);

блоки УСО — функционально законченные устройства, включающие в себя все необходимые схемы преобразования сигналов, узлы гальванического разделения и средства взрывозащиты без вынесения этих функций в кроссовые секции;

каждое направление ИР имеет шину аналогового сигнала, поэтому блоки приема аналоговых сигналов содержат только коммутаторы, а АЦП — один на всю магистраль в контроллере;

каждый сигнал ИР занимает два параллельных контакта в разъеме интерфейса, что повышает устойчивость к отказам по вине разъемных соединений;

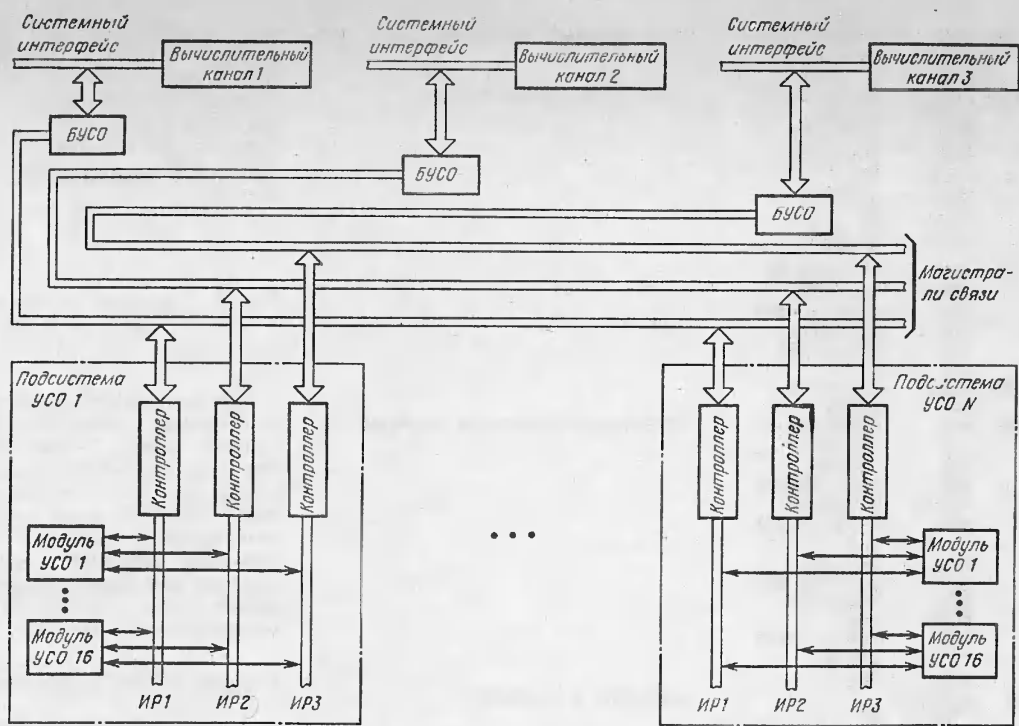


Рис. 1. Структура организации подключения подсистем УСО в ПС1001

блоки вывода и радиальный сигнал Выборка* имеют обратную связь, поэтому могут быть проконтролированы (надежность и достоверность информации, выдаваемой на объект на ИР увеличивается);

предусмотрена схема формирования контрольных напряжений с их последующим контролем во всех блоках УСФ (диагностируется несанкционированный выход на магистральные шины данных и управления).

Сигнал ИР по отношению к блоку УСО

- На радиальной шине:
 - входной — ВБР (Выборка)
 - выходной — ПР (Запрос на прерывание)
- На магистральной шине:
 - входные — ЗД (Запрос данных)
 - НД (Наличие данных)
 - РАБ (Работоспособность)
 - выходные — ПВБР (Подтверждение выборки)
 - ПОП (Подтверждение операций)
 - ЗАП (Запуск АЦП)
- На двунаправленных шинах:
 - Д00...Д15 — 16 сигналов данных.

Обмен информацией по шинам каждого из трех направлений ИР осуществляется по асинхронному принципу запрос — ответ.

Рассмотрим алгоритм обмена по одному направлению ИР. В одном резервированном управляющем вычислительном комплексе в данный момент времени может работать от одного до трех вычислительных каналов или центральных процессоров. Каждый исправный вычислительный канал формирует серию импульсов, которая через магистраль связи поступает на одновibrator с повторным запуском, расположенный в контроллере данного направления и подающий на него сигнал РАБ. При отказе канала серия импульсов прерывается и сигнал РАБ снимается. Триггеры и регистры данного направления сбрасываются, а входные и выходные сигналы ИР устанавливаются в состояние высокого

импеданса. В блоках УСО при отсутствии сигнала РАБ (по предварительной договоренности) либо предыдущее состояние запоминается и поддерживается на выходных элементах, предназначенных для управления объектом при исчезновении РАБ, либо выходные элементы данного направления устанавливаются в заданное предпочтительное состояние. По сигналу РАБ в случае необходимости осуществляется программная организация сигнала «Сброс» по данному направлению, а конъюнкцией РАБ всех трех направлений — сигнала «Общий сброс».

Сигнал ВБР вырабатывается контроллером и поступает на соответствующую радиальную шину по полученному из центра адресу блока УСО, к которому производится обращение в виде 4-разрядного двоичного кода. Каждый блок УСО, получив сигнал ВБР, обязан выдать сигнал ПВБР и сохранять его на все время действия ВБР. Наличие этих двух сигналов на шинах ИР — неперемное условие возможности обмена информацией между выбранным блоком УСО и контроллером.

Двунаправленный обмен информацией осуществляется по шинам данных Д00...Д15: от контроллера к блоку в сопровождении сигнала НД (рис. 2, а) и от блока к контроллеру сигнала ЗД (рис. 2, б). Сигналы НД и ЗД формируются контроллером; квитанция их принятия — сигнал ПОП, с помощью которого блок ввода сопровождает информацию на шинах Д00...Д15 и блок вывода сообщает о приеме информации, переданной контроллером. Кроме этого, по шинам Д00...Д15 в сопровождении сигнала ВБР контроллер выдает многоканальному блоку УСО адрес канала, с которым будет производиться обмен информацией. В отличие от сигнала ВБР, не запоминаемого адресованным блоком, адрес канала поступившего по шинам Д00...Д15, блок должен запомнить, так как после получения сигнала ПВБР шины Д00...Д15 освобождаются контроллером для последующего обмена данными. Адресованному блоку УСО в сопровождении сигнала ВБР по шинам Д00...Д15 передается код осуществляемой контрольной функции.

Инициативные блоки УСО выдают сигналы ПР, посту-

* А. С. 1361529 СССР, МКИГОГФ3100. Устройство для сопряжения процессора с абонентами / Н. И. Гайдашенко — Оpubл. 1987, Бюл. № 41.

пающие по радиальным шинам в контроллер, который осуществляет дизъюнкцию всех 16 (по числу посадочных мест в каркасе для блоков УСО) сигналов ПР и подает в центр обобщенный сигнал запроса прерывания. Центральный процессор в первую очередь выполняет запрос вектора инициативы, который хранится в контроллере в виде 16-разрядного позиционного кода. Сигнал ПР, проявивший инициативу, снимается блоком УСО опросом данного блока по вышеописанному алгоритму.

Блок ввода аналоговых сигналов работает с групповым АЦП, расположенным в контроллере. В этом случае шины данных Д00...Д15 не используются и вместо сигнала ПВБР формируется сигнал ЗАП (рис. 2, в), запускающий АЦП с задержкой на время открывания аналоговых ключей. Контроллер сообщает результат аналого-цифрового преобразования центральному процессору и снимает сигнал ВБР, прекращая операции по шинам интерфейса ИР.

Таким образом пассивный контроллер транслирует в блоки УСО управляющие слова, получаемые от процессора, слова данных. Он обеспечивает индивидуальное обращение к любому входному или выходному каналу связи с объектом. В резервированных комплексах блок ввода УСО выдает информацию по запросу на любое из трех направлений ИР, а в синхронном режиме — на все три направления практически одновременно (при наличии трех запросов). В синхронном режиме блок вывода УСО получает информацию одновременно с трех направлений и преобразует ее по мажоритарному принципу (в случае аналогового вывода — по принципу медианы); на объект поступает наиболее достоверный сигнал.

Активный контроллер с подключенными через ИР блоками УСО самостоятельный элемент децентрализованной структуры. Его функции:

- первичная обработка входной информации;
- контроль по заданным уставкам;
- фиксация моментов появления инициативных сигналов;
- реализация компенсации температуры холодных сплавов (для ввода сигналов от термопар);
- подсчет числа импульсов, измерение частоты и длительности импульсов с контролем по заданным уставкам;
- выдача числа или длительности с установкой параметра по обратной связи;
- непосредственное цифровое или логическое управление.

Одни и те же блоки УСО, конструктивно выполненные в виде функционально законченных устройств, могут ис-

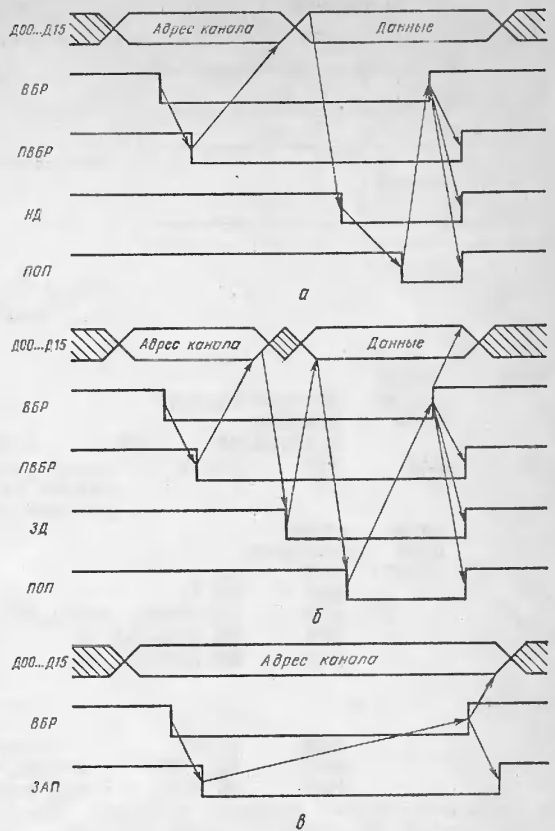


Рис. 2. Диаграммы сигналов магистрали ИР: а — запись данных в блоки УСО; б — чтение данных из блоков УСО; в — цикл обмена для ввода аналоговых сигналов

пользоваться в системах как с активным, так и с пассивным контроллерами.

349940, Северодонецк-5, НПО «Импульс»; тел. 90-5-97

Статья поступила 15.03.88

УДК 681.069

И. П. Стацук, А. П. Александрович, А. В. Захаревич

АППАРАТНО-ПРОГРАММНЫЙ ИНТЕРФЕЙС ДЛЯ ОБМЕНА ИНФОРМАЦИЕЙ МЕЖДУ ПЭВМ ЕС1840 И ДВКЗ

Связь организована на базе последовательного интерфейса RS232 ЕС1840 и интерфейса радикального последовательного (ИРПС) ДВКЗ. Схемное решение выходных усилителей используемых цепей интерфейса RS232 показано на рис. 1. Способ соединения цепей ИРПС и RS232 дан на рис. 2. Задействованы только пять цепей RS232:

103 — передаваемые данные; соединяется с цепью ПР (принимаемые данные) ИРПС;

104 — принимаемые данные — с цепью ПД (передаваемые данные) ИРПС;

106 — готовность принять данные — с цепью ГПР (готовность приемника) ИРПС;

107 — готовность передать данные; заведена на схемную землю (Лог. 1);

108 — запрос о готовности передать данные — с цепью ГПД (готовность передатчика) ИРПС.

В режиме приема посылки от КТЛК

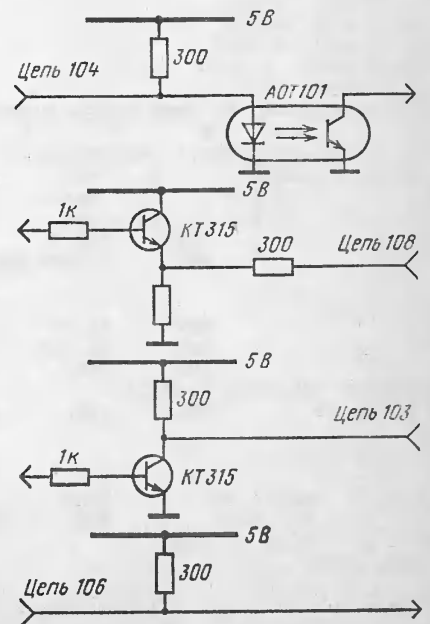


Рис. 1. Доработка выходных усилителей цепей стыка С2

RS232	ИРПС
Цепь 103	ПР
Цепь 104	ПД
Цепь 106	ГПР
Цепь 107	
Цепь 108	ГПД

Рис. 2. Схема соединения цепей ИРПС и стыка С2

Рис. 3. Программа-обработчик прерывания INT14

```

NAME      INT14
ASSUME    CS:CODE,DS:DATA
DATA      SEGMENT
RS232_BASE DW      4 DUP(?)
DAT       EQU      0040h ; адрес системной области
                                ; данных для работы с
                                ; портами ввода-вывода

DATA      ENDS
CODE      SEGMENT
FIRST:    SUB       AX,AX
          MOV       ES,AX
          MOV       AX,OFFSET RS232_IO ; заносим адрес
          MOV       ES:[0050h],AX    ; точки входа в
          MOV       ES:[0052h],CS    ; обработчик в
          ; вектор обра-
          ; ботки прерыва-
          ; ния INT 14
          SUB       DX,DX ; адаптер 0
          MOV       AH,0C8h ; инициализация портов
          MOV       AL,0CAh ; под параметры линии
; Параметры линии: скорость передачи, контроль паритета,
; количество стоп-битов, длина посылки, заданные положе-
; нием перемычек на плате КТЛК ДВКЗ . В данном случае
; последовательный порт 0 инициализируется со следующими
; параметрами:
; скорость - 4800 Бод, стоп-битов - 2
; длина - 8 бит, нет контроля паритета
;
; входные данные (обычный формат):
;
;   7 6 5 4 3 2 1 0
;   |-----| |-----| |-----| |-----|
;   | скорость | 1 0 - функция
;   | 1-50, 2-75 | 00-инициализация
;   | 3-100, 4-150 |
;   | 5-200, 6-300 |
;   | 7-600, 8-1200 |
;   | 9-2400, A-4800 |
;   | C-9600 |
;
;   7 6 5 4 3 2 1 0
;   |-----| |-----| |-----| |-----|
;   | стоп-бит | | контроль | | длина | |
;   | 01-1 | X0 - чет | 10 - 7 бит | 1
;   | 10-1.5 | 01 - нечет | 11 - 8 бит | 0
;   | 11-2 | 11 - чет |
;
          INT      14h
          MOV     DX,BUF_SIZE ; количество параграфов
                                ; для размещения задачи
                                ; при инсталлировании

          MOV     AL,00
          MOV     AH,31h
          INT     21h ; оставить резидентной

ASSUME    CS:CODE,DS:DATA
RS232_IO PROC FAR ; точка входа в обработчик
                . ; обработчик прерывания
                . ; INT 14 BIOS IBM PC
                .
                .
                .
          RS232_IO ENDP
          BUF_SIZE EQU ($-OFFSET FIRST)/16+1
          CODE     ENDS
END FIRST

```

ДВКЗ со стороны ПЭВМ ЕС1840 выполняются следующие действия: установка цепи 108 RS232, что указывает на готовность приема данных, ожидание и прием посылки, сброс запроса по цепи 108. В режиме передачи посылки — ожидание готовности КТЛК принять посылку (Лог. 1 в цепи 106), готовность КТЛК к передаче посылки, прием данных в буфер КТЛК, сброс готовности КТЛК принимать данные (Лог. 0 в цепи 106). Со стороны ДВКЗ выполняются стандартные действия с регистрами состояния и буферными регистрами приемника и передатчика.

Интерфейс позволяет организовать прием и передачу посылок между ДВКЗ и ПЭВМ ЕС1840, со скоростями 50, 75, 100, 150, 200, 300, 600, 1200, 2400, 4800, 9600 Бод в форматах семи или восьми информационных битов, а также формирование и контроль бита паритета или работу без него. Конкретные технические характеристики ДВКЗ задаются аппаратно установкой перемычек на плате КТЛК, а ПЭВМ ЕС1840 — программированием микросхем платы адаптера интерфейсов. Большинство прикладных и системных программ, работающих с интерфейсом RS232, используют для этой цели прерывание INT14, поддерживаемое BIOS. В BIOS ПЭВМ ЕС1840 обработчик прерывания INT14 отсутствует, вместо него установлена «заглушка». Поэтому возникла необходимость в написании специальной программы (рис. 3) на основе стандартного обработчика прерывания INT14 из BIOS IBM PC*. Программа заносится в память машины во время загрузки ОС запуском из командного файла AUTOEXEC.BAT. В остальной работа с интерфейсом RS232 осуществляется стандартно.

Помехозащищенность обмена данными на уровне звена связи между ДВКЗ и ПЭВМ ЕС1840 обеспечивается с помощью стандартного протокола канала передачи данных стандарта SECS, рекомендованного МККТТ в качестве стандарта для управления технологическим оборудованием. Стандарт SECS не рассчитан на передачу больших массивов данных, но повышает надежность передачи данных в сложной помеховой обстановке. Передаваемые блоки информации могут содержать как двоичные знаки BINARY, так и стандартные ASCII символы.

220600, Минск, ул. П. Бровки, 6. МРТИ,
каф. ТР5А; тел. 39-88-35

Статья поступила 9.03.89

* Шнайдер А. Язык ассемблера для персонального компьютера фирмы IBM / Пер. с англ. — М.: Мир, 1988. — 406 с.

Д. А. Лучук

СИНХРОННО-АСИНХРОННЫЙ АДАПТЕР С ВЫХОДОМ НА СТЫК С2 ДЛЯ ЭВМ ТИПА «ЭЛЕКТРОНИКА 60»

Синхронно-асинхронный адаптер (САА) выполнен на плате конструктива микроЭВМ «Электроника 60» и обеспечивает подключение одного канала обмена данными, которое может осуществляться в дуплексном или полудуплексном режиме со стандартными скоростями. Обмен с системной магистралью микроЭВМ — программный или по прерыванию.

В основе работы САА (см. рисунок) — принцип преобразования параллельного кода, принятого из системной магистрали ЭВМ, в

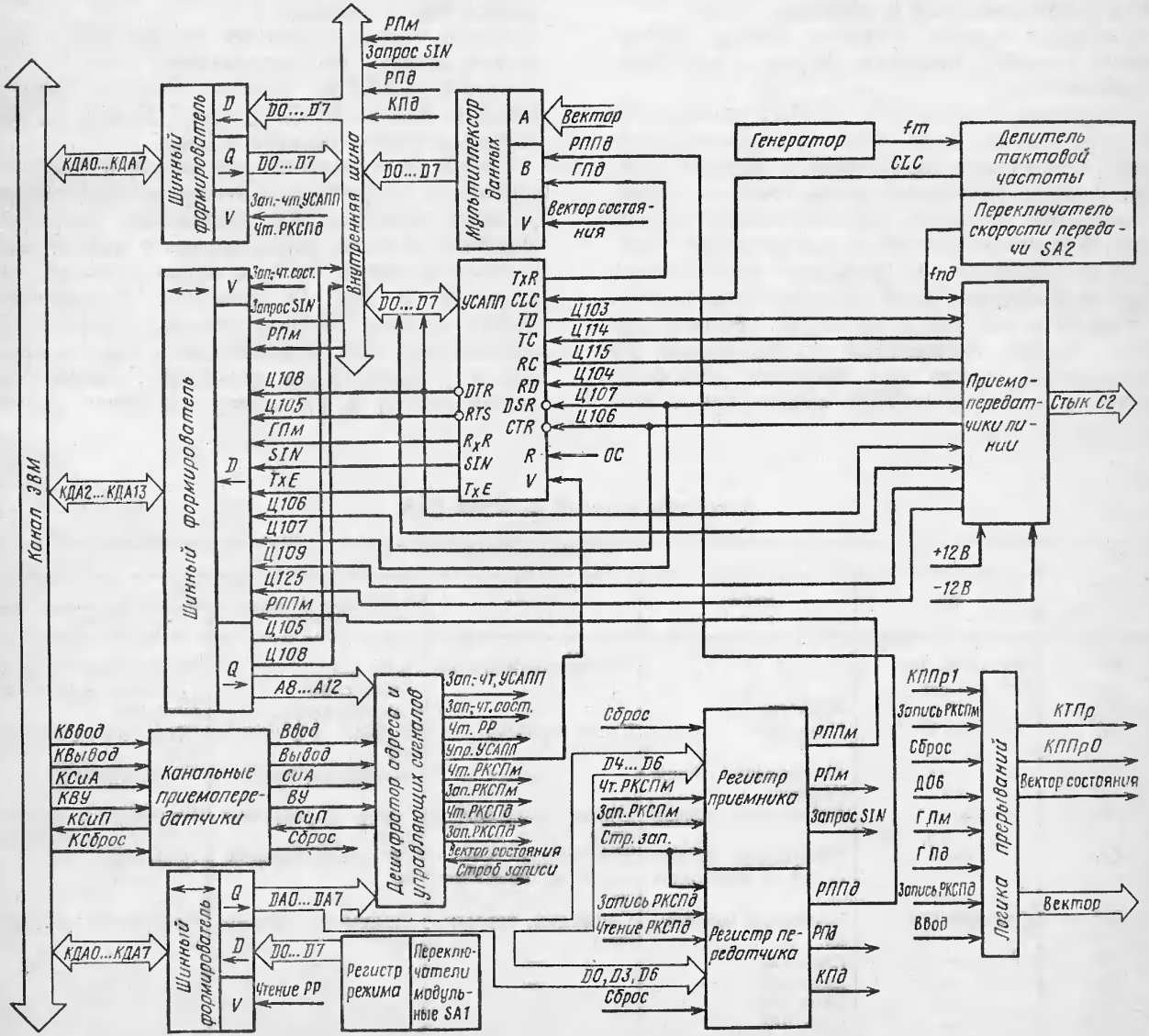
последовательный и его трансляция по двупроводной линии аппаратуры передачи данных (АПД), а также обратного преобразования и передачи в ЭВМ. В САА реализованы только основные цепи стыка С2 (табл. 1). В качестве центрального элемента применена БИС универсального синхронно-асинхронного приемопередатчика (УСАПП) типа К580ИК51 для увеличения степени интеграции и надежности, а также уменьшения габаритов САА.

Функции УСАПП:

установка режима работы в соответствии с инструкцией;

разрешение приемопередачи в полудуплексном либо приема и передачи в дуплексном режиме;

поиск знаков синхронизации SIN; выключение и включение цепей стыка С2 в соответствии с командной инструкцией;



Структурная схема САА

Таблица 1

Цепи стыка С2

Цепь	Международное обозначение	Назначение
103	TD	Передаваемые данные
104	RD	Принимаемые данные
105	RTS	Запрос передачи
106	CTS	Готов к передаче
107	DSR	АПД готова
108	DTR	Подключить АПД к линии
109	CD	Детектор принимаемого линейного сигнала
114	TC	Синхронизация элементов передаваемого сигнала
115	RC	Синхронизация элементов принимаемого сигнала
125	Ring	Индикатор вызова

преобразование данных последовательного вида в параллельный и обратно;

фиксация ошибок процесса приема информации (ошибки паритета, формата или переполнения).

Процессор к регистрам УСАПП обращается по внутренней шине САА и двунаправленному восьмиразрядному каналу данных БИС (применение внутренней шины связано с необходимостью введения дополнительных регистров). Это предоставляет дополнительные удобства пользователю и позволяет использовать драйверы ввода-вывода операционных систем.

Работу в режиме прерываний обеспечивает узел логики прерываний, выполненный по стандартной схеме для внешних устройств, работающих с микроЭВМ «Электроника 60».

Адрес вектора прерывания передается в системную магистраль микроЭВМ через мультиплексор данных, который управляется сигналом вектора состояния.

Дешифратор адреса и управляющих сигналов определяет режим работы САА. В САА пять адресуемых регистров: РД — регистр данных, РКСПм — регистр команд и состояний приемника, РКСПд — регистр команд и состояний передатчика, РР — регистр режима, РКС УСАПП — регистр команд и состояний УСАПП. Назначения разрядов остальных регистров приведены в табл. 2.

В зависимости от состояния адресных разрядов А0, А1, А2 вырабатываются внутренние сигналы управления устройством:

запись-чтение УСАПП и управление УСААПП — при чтении регистров УСАПП, записи инструкций режима и команд, чтении-записи байта данных;

запись-чтение состояния, чтение РР — при работе шинных формирователей;

чтение РКСПм, запись РКСПм, чтение РКСПд, запись РКСПд, строб записи — при записи и чтении регистров САА.

Состояние триггеров для записи инструкций прочитать невозможно, поэтому при программировании режима САА необходимо прочитать значение режима, установленного модульными переключателями SA1, а затем записать это значение в РР (в процессе работы это всегда можно сделать, так как сохраняется соответствие между положением переключателей и установленным режимом). Знаки SIN записываются в РР сразу же после записи

Назначения разрядов регистров САА

Таблица 2

Разряд.	Регистры САА				
	РД	РКСПм	РКСПд	РКС УСАПП	
				Чтение	Запись
D0	Младший бит	—	Разрешение передачи	Цепь 107	Разрешение поиска
D1	Бит данных	—	—	Обнаружен	Установка в «0»
D2	»—»	Цепь 105	—	Ошибка кадра	Цепь 105
D3	»—»	Цепь 108	Конец передачи	Ошибка переполнения	Сброс триггеров ошибок
D4	»—»	Разрешение поиска SIN	—	Ошибка паритета	Конец передачи
D5	»—»	Разрешение приема	Буфер передатчика пуст	Буфер передатчика пуст	Разрешение приема
D6	»—»	Разрешение прерывания от приемника	Разрешение прерывания от передатчика	Готовность приемника	Цепь 108
D7	Старший бит	Готовность приемника	Готовность передатчика	Готовность передатчика	Разрешение передачи
D8	—	Цепь 125	—	—	—
D9	—	Цепь 109	—	—	—
D10	—	Цепь 106	—	—	—
D11	—	Цепь 107	—	—	—
D12	—	SIN обнаружен	—	—	—

инструкции синхронного режима работы и хранятся в УСАПП. Чтение записанных знаков SIN невозможно.

Информация, записанная в САА после его установки в исходное состояние, будет воспринята как инструкция режима.

При передаче в асинхронном режиме инструкция команды, содержащая указание о включении цепей 108 и 105, записывается либо через РКС УСАПП, либо через РКСПм и РКСПд (при работе в режиме прерывания программы). Если цепи 108 и 105 включены, то цепи 106, 107 переходят в активное состояние и на входах DSP, CTS УСАПП появляются низкие уровни сигналов. Получив сообщения о готовности модема и передатчика, фиксируемые в регистрах состояний, УСАПП устанавливает разряды D0 в РКС УСАПП и D7 в РКСПд в единичное состояние, разрешая передачу.

Процессор, проверив готовность устройства либо прервав выполнение программы в режиме прерываний, записывает первый байт данных в РД. Данные будут выданы в последовательном виде с установленными скоростью и форматом на выход TD УСАПП и через передатчик в цепь 103, если передача разрешена командной инструкцией и цепь 106 сохраняет состояние «Включено».

Устройство принимает данные, поступающие на вход RD УСАПП из цепи 104. Если прием разрешен и знак собран, разряды D1 и D7 переходят в активное состояние. Одновременно в РКС УСАПП фиксируются ошибки, которые могли возникнуть при приеме. Процессор считывает байт данных из регистра данных приемника, после чего разряды готовности сбрасываются.

Передачи в синхронном и асинхронном режимах принципиально не различаются, однако в синхронном режиме в посылках нет стартовых и стоповых битов, а перед массивом знаков данных идут SIN1 и SIN2.

Знак SIN2 и бит контроля могут отсутствовать (SIN записываются в PP непосредственно после инструкции синхронного режима). Выход TD УСАПП имеет состояние Log.1 до загрузки байта данных в буферный регистр передатчика. Этот байт данных может принимать любое значение, так как в процессе поиска синхросимволов приемником АПД будет утерян. Целесообразно код первого байта устанавливать аналогично коду синхросимвола.

Если цепь 106 включена и передача разрешена, данные побитно передаются с выхода TD через передатчик в цепь 103. Скорость передачи данных определяется серией тактовых импульсов, подаваемых на вход TC УСАПП с приемника цепи 114. Если процессор не записывает в буферный регистр передатчика

байт данных, то УСАПП передает SIN по цепи 103.

При приеме информации УСАПП сравнивает коды байтов цепи 104 с кодами запрограммированных синхросимволов, если разрешены прием и поиск символов синхронизации командной инструкцией. Если принятый и заданный знаки одинаковы, устройство переходит в режим синхронизации. Скорость принимаемых данных определяется частотой тактовых импульсов, поступающих с приемника цепи 115 на вход RC УСАПП.

В работе с устройством надо учитывать некоторые особенности: при записи информации в устройство необходим интервал длительностью не менее шести тактов синхронизации, подаваемых на вход CLC УСАПП, а после программного сброса перед записью инструкции режима — задержка не менее 15 тактов синхронизации.

Для исключения ошибок при ложном поступлении сигнала готовности УСАПП (вывод R_R) нужно после программирования устройства на прием информации прочитать регистр данных приемника без учета результата (ложный сигнал готовности сбрасывается).

В модуле САА использованы распространенные микросхемы серии K155, K531, K559, K170. С его помощью в систему можно подключить любое оборудование, имеющее выход на стык С2.

252207, Киев, пр. акад. Глушкова, 20,
ИК АН УССР; тел. 265-83-95

Статья поступила 2.02.88

КРАТКИЕ СООБЩЕНИЯ

УДК 681.3.06

С. И. Горшенев

ДРАЙВЕР ДЛЯ ОРГАНИЗАЦИИ МЕЖМАШИНОЙ СВЯЗИ

Драйвер предназначен для программной поддержки связи микроЭВМ «Электроника 60» и мини-ЭВМ SM4 через четырехпроводную линию с помощью устройства последовательного обмена (УПО) и блока дистанционной связи SM8502 (БС АДС).

Драйвер управляет работой УПО микроЭВМ «Электроника 60» в ОС RT-11 версии 5.0, обслуживает запросы приема и передачи информации по каналу межмашинной связи, имеет имя UP.SYS, занимает на диске два блока, а в памяти 250 слов. Стандартные для ОС RT-11 запросы на открытие канала, чтения или записи позволяют выполнять прием и передачу данных, например макрокомандами или с использованием подпрограмм системной объектной библиотеки на языке Фортран.

Сложность организации ввода-вывода в такой системе связана с синхронизацией обмена. В БС АДС и в УПО отсутствуют аппаратные средства, по которым можно было бы определить установленные запросы ввода-вывода в ЭВМ на другом конце линии связи, а также отследить ошибки при обмене. Проблему можно частично

решить, рассматривая БС АДС в СМ4 как один из терминалов, что оговаривается при генерации ОС RSX-11M. В этом случае незапланированный ввод с линии связи в СМ4 воспринимается как команды монитору ОС, что позволяет программе, работающей в «Электронике 60», зарегистрироваться в качестве пользователя в СМ4 и запустить задачу межмашинного обмена.

После регистрации обмен инициализируется командой запуска программы обмена в СМ4. Для приема данных из СМ4 (после передачи команды запуска программ по запросу WRITE) достаточно установить запрос READ. Однако для передачи данных в СМ4 вначале следует убедиться в готовности программы к приему. Для этой цели в драйвере предусмотрено обслуживание запроса с номером 367 по специальной форме.

При использовании запросов чтения программа должна следить за «зависанием», которое может быть обусловлено ошибками приема-передачи. Снять запросы можно с помощью макрокоманды АВТЮ, однако при выполнении программ обработки прерываний это неудобно, так как требует предварительного выполнения запросов INTEN и SYNCH. Поэтому в драйвере предусмотрено снятие всех запросов командным прерыванием IOT. В этом случае в системе фиксируется ошибка на устройстве, которая снимается следующим запросом ввода-вывода или макрокомандой WAIT.

Телефон 551-75-47, Москва

Сообщение поступило 5.12.88

УДК 681.3.06

О. П. Мосоян, С. М. Меликян, Н. В. Огонесян

СЕТЕВОЕ ПО НА БАЗЕ ОС РВ И РАФОС 2

На базе ОС РВ и РАФОС 2 разработано сетевое программное обеспечение (СПО) для построения радиальной структуры. В качестве базовой ОС центральной машины (ЦМ) используется ОС РВ, периферийной машины (ПМ) — РАФОС, предназначенная для организации высокопроизводительных систем с разделением функций. СПО не требует изменений и дополнений программных средств ПМ, не зависит от версии ОС РВ и конкретной генерации, что обеспечивает возможность его совместного использования с другим СПО, позволяет построить ЛВС с ПМ, функционирующими под управлением различных ОС.

СПО обеспечивает два режима работы ПМ: удаленного терминала и доступа к файлам ЦМ. В режиме 1 пользователь ПМ имеет доступ к ОС РВ и непосредственный контакт с ее управляющей программой. В режиме 2 пользователю ПМ предоставляются возможности ОС РАФОС с доступом к дискам ЦМ как к собственным дискам ПМ.

В состав СПО входят драйвер линии связи, загрузчик ОС ПМ, процессор удаленных файлов (ПУФ).

Драйвер осуществляет инициализацию, передачу-прием по каналам, отменяет ввод-вывод, отключает линию связи.

Загрузчик ОС активизирует драйвер и запускает ПУФ. Процесс загрузки продолжает ПУФ, обеспечивающий доступ к файлам на дисковых носителях ЦМ из системных и пользовательских задач ПМ. Алгоритм работы ПУФ аналогичен алгоритму программы DISK, функционирующей в среде РАФОС 2. Только в отличие от DISK в ПУФ предусмотрен механизм установления соответствия файловых структур РС РВ и РАФОС. В этих целях в ПУФ создан виртуальный диск (ВД) формата RT. Соответствие поддерживается с помощью команд оператора ЦМ или ПМ в режиме удаленного терминала. Каталог ВД запоминается в области данных ПУФ.

Адресации к различным ВД распознаются ПУФ по номерам логических каналов, подсоединенных к ВД со стороны ПМ. Запросы обслуживаются путем их распознавания и выполнения соответствующих операций в среде ОС РВ. ПУФ реализует следующие типы запросов ПМ: чтение и

запись блока ВД, передачу-прием сообщения, монтировку ВД.

Все файловые операции, адресованные к ВД, поступают в ПУФ как последовательные комбинации запросов на чтение и (или) запись блока ВД. Различают блоки каталога и файла ВД.

Запрос на чтение виртуального блока каталога обслуживается без обращения к дискам ЦМ: выбирается часть сформированного каталога ВД и передается в линию связи. Если поступил запрос на запись, то блок, полученный через линию связи, сравнивается с соответствующим блоком каталога ВД. Различия между ними определяют необходимую функцию для фактической реализации данной операции — создания или удаления файла. После этого старый блок каталога ВД заменяется новым, принятым из линии связи.

Если запрос относится к файлам ВД, то по номеру блока, полученного в блоке запроса, определяется, к какому файлу адресован запрос, и выполняется необходимая функция — чтение или запись.

При обслуживании запросов на монтировку ВД ПУФ пользуется таблицей соответствий ВД и каталогов тома ЦМ. ПУФ определяет требуемый ВД, составляет его каталог и тем самым обеспечивает доступ из ПМ для дальнейших операций. Помимо файловых операций ПУФ осуществляет функции, связанные с ВД в целом, — инициализацию и копирование ВД.

СПО может работать на СМ4 стандартной конфигурации с минимальной оперативной памятью 128 Кбайт в качестве ЦМ и на микроЭВМ «Электроника 60» с оперативной памятью 32 Кбайт в качестве ПМ.

Сетевые интерфейсы могут служить ИРПС, БС АДС (блок системного адаптера дистанционной связи), БИ (блок интерфейсный), УПО (устройство последовательного обмена), мультиплексор МПА-1.

Телефон 35-48-14, Ереван

Сообщение поступило 21.04.88

УДК 681.3.06

Е. А. Коноплев, С. В. Токарев

ДРАЙВЕР ПЕЧАТАЮЩЕГО УСТРОЙСТВА МОЗАИЧНОГО ТИПА

Разработан драйвер печатающего устройства (ПУ), который при установке SET-параметров позволяет максимально реализовать все возможности, заложенные в аппаратных средствах ПУ.

Для широко распространенного ПУ D-100 (EC7189, SM6325) он обеспечивает:

работу со всеми шрифтами (русским, латинским, строчным, прописным);

перекодировку шрифтов (строчных в прописные, латинских в русские и наоборот);

выделение текста или его части несколькими способами (двойной пропечаткой, печатью с фазовым сдвигом, узкой печатью, печатью удвоенной высоты и/или ширины);

отступление текста слева на N позиций от начала страницы и регулирование числа символов в строке;

возможность смены параметров драйверов CSR, ESC, VECTOR, SYSGEN и получения «твердой копии» содержимого графического ОЗУ (платы графики КГД, КЦГД, УКНЦ);

работу со страницами (их нумерацию, изменение числа строк на странице и пустых строк между ними, смену строчный между строками, запрос на печать новой страницы, обработку символа FF).

При загрузке ОС (RT-11 не ниже четвертой версии) драйвер автоматически заносится в настроенную таблицу. Он занимает четыре блока дискового пространства и 670 слов оперативной памяти в загруженном состоянии.

При работе ПУ с ограниченными (по сравнению с D-100) аппаратными возможностями ресурсы конкретного ПУ благодаря драйверу используются максимально

Сообщение поступило 22.03.89

Тел. 484-95-17, Киев

УДК 681.322.042

Р. И. Грушвицкий, А. В. Преображенский, А. В. Симонов, В. М. Яккер

КЛАСС УЧЕБНЫХ ЭВМ В ЛАБОРАТОРНОМ ПРАКТИКУМЕ «ОПЕРАЦИОННЫЕ СИСТЕМЫ»

Знание структуры, принципов организации и функционирования, а также возможностей, предоставляемых различными ОС, необходимо для эффективного и рационального использования ресурсов вычислительных систем. Особую актуальность эти вопросы приобрели в связи с широким внедрением персональных ЭВМ (ПЭВМ) в сферу деятельности многочисленной группы непрофессиональных пользователей. Важнейшая проблема — быстрое и целенаправленное обучение пользователей основам организации ОС микро- и ПЭВМ, использованию системных средств.

Основные задачи лабораторного практикума

Рассматривая ОС как многоуровневую систему, выделяем в ее составе три основных иерархически взаимосвязанных и вложенных друг в друга уровня, включающих в себя соответственно:

средства поддержания диалога — интерпретатор командного языка;

унифицированные средства поддержания протоколов доступа (на логическом уровне) к подчиненным ОС ресурсам;

средства взаимодействия в конкретной аппаратной конфигурации со стандартным набором физических ресурсов.

Вопросы реализации взаимодействия со стандартными ресурсами на физическом уровне практического интереса для большинства пользователей не представляют. По мере совершенствования методов и средств организации диалога изучение уровня командного языка все более упрощается. Центральной задачей практикума становится изучение структуры ОС и протоколов доступа к ресурсам системы на логическом уровне.

Для изучения выбрана распространенная ОС CP/M80. Структура и основные принципы функционирования ее просты и наглядны в дальнейшем развитии в ряде более сложных ОС — 16-разрядных ПЭВМ (MS DOS [1]).

Техническое обеспечение учебного класса

Для реализации архитектуры учебного класса, совмещающей высокую степень использования оборудования и простоту средств доступа обучающихся к системным ресурсам, выбрана архитектура локальной сети (ЛС) с иерархически-звездообразной топологией, центральным управлением, витой парой проводов в качестве среды передачи данных. Для доступа к ресурсам коллективного пользования применены специализированные алгоритмы.

Введение в архитектуру дополнительного уровня файловой памяти — электронного имитатора диска (ЭИД) существенно увеличило производительность, надежность и экономичность ЛС. В зависимости от места размещения и способа реализации функции такого ЭИД могут быть различны:

- 1) ОЗУ ЭИД, входящий в состав персонального рабочего места (ПРМ) для снижения нагрузки на систему передачи данных;
- 2) ПЗУ ЭИД, включаемый в состав ПРМ в виде сменной кассеты для автономной работы;
- 3) ОЗУ ЭИД, входящий в состав центральной вычислительной системы (ЦВС) и для разгрузки НМД от обращений за наиболее

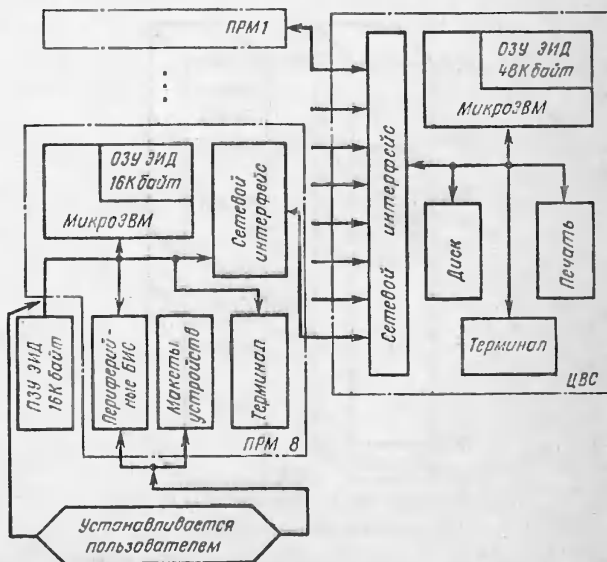


Рис. 1. Структура аппаратных средств учебной лаборатории

часто используемыми программными средствами;

4) ПЗУ ЭИД, выполняющий в ЦВС те же функции, что и ОЗУ ЭИД, однако допускающий исключение НМД из состава ЦВС.

Структура ЛС (рис. 1) с дополнительным уровнем файловой памяти позволяет легко адаптировать сеть к решению различных задач в конкретных условиях разработки и применения. Аппаратура ЛС включает в себя:

8 ПРМ, имеющих ОЗУ ЭИД (16 Кбайт) и допускающих внешнее подключение ПЗУ ЭИД такого же объема [2];

систему передачи данных, образованную реальными каналами связи, использующими последовательные приемопередатчики КР580ИК51А, работающие в асинхронном режиме на скорости 9,6 КБод;

ЦВС, построенную, как и ПРМ, на базе БИС серии К580 и имеющую в своем составе ОЗУ ЭИД (48 Кбайт) и периферийное оборудование: НГМД (2×254 Кбайт) и печатающее устройство.

Построить систему передачи данных [3] довольно просто, однако аппаратно-программная структура ПРМ и ЦВС содержит ряд нестандартных решений, связанных с введением ЭИД.

Наиболее сложно оказалось создать гибкое ПО, поддерживающее эффективное функционирование системы. Особенность ПО ЛС — введение в состав резидентного ПО ПРМ (кроме отладочного монитора и сетевого загрузчика) образа ядра ОС СР/М, размещаемого в ПЗУ, для автономной работы ПРМ. Таким образом, обучающемуся предоставляется ПРМ с виртуальной структурой, соответствующей развитой ПЭВМ, и обеспечивается высокая надежность занятий.

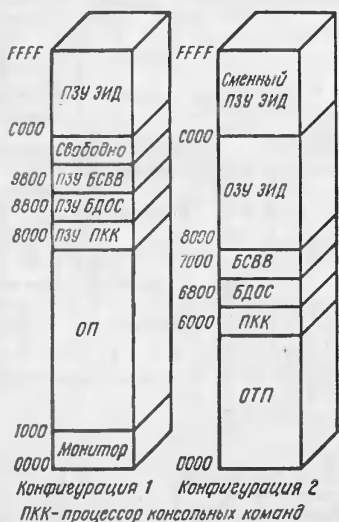


Рис. 2. Схема распределения памяти персонального рабочего места для действующей версии ПО

В текущей версии ПО используются две конфигурации памяти ПРМ (рис. 2). При этом выделим ряд состояний ПРМ с разными возможностями для обучающихся:

1. После включения питания управление передается монитору для разработки и отладки программы непосредственно в кодах МП. Память имеет конфигурацию 1.

2. При введении директивы загрузки ОС управление передается загрузчику. Память переключается на конфигурацию 2; загружается и запускается ядро СР/М.

3. Базовая система ввода-вывода (БСВВ) проверяет факт подключения ПРМ к ЛС. Если после 500 мс ответ не приходит, то обучающийся получает сообщение о работе ПРМ в автономном режиме и о доступных ресурсах: внутреннем ОЗУ ЭИД (А:) и внешнем ПЗУ ЭИД (В:).

4. Получив ответ от сети, ПРМ автоматически переходит в сетевой режим: во-первых, на основании имени, содержащегося в запросе, обучающемуся выделяется область на диске (в версии ПО — 30 Кбайт); во-вторых, обучающемуся сообщается о факте подключения к ЛС и список доступных ресурсов — локальные диски А: и В:, ОЗУ ЭИД коллективного пользования, содержащее набор наиболее часто используемых программных средств (С:), личный диск (Д:) и печатающее устройство (РРН:).

Два режима работы резидентной ОС ПРМ (автономный и сетевой) резко увеличивают надежность проведения лабораторных работ. Однако отметим, что по предоставляемым методическим возможностям и эффективности процесса обучения автономный режим существенно уступает сетевому.

При работе ПРМ в автономном режиме основной носитель системных и учебных программных средств — сменный ПЗУ ЭИД. Разработаны следующие комплекты сменных кассет-ЭИД:

интерактивный отладчик DDT с комплектом учебных программ;

учебный интерпретатор языка БЕЙСИК, расширенный функциями по управлению вводом-выводом и другими;

интерпретатор языка ФОРТ;

учебная система управления базами данных.

Перспективно развить предложенный подход к построению учебных ЛС. Так, функциональные возможности сети можно существенно расширить при увеличении объемов ОПП как ПРМ, так и ЦВС. Увеличение объема сменных ПЗУ ЭИД расширит функциональные возможности ПРМ в автономном режиме. И наконец, применение в качестве ЭИД модулей энергонезависимых ОЗУ сохранит файлы обучающихся между занятиями при автоном-

ной работе и в случае отказов сети. Это повысит надежность работы системы в целом.

Методика проведения занятий

При разработке методики выделены принципы, обеспечивающие целостность и эффективность курса. К ним относятся: нарастающая сложность и преемственность работ, активное использование демонстрационных программ, сочетание индивидуального и коллективного участия в разработке программы для развития навыков программирования крупных задач, максимальная приближенность учебных задач к реальным.

Для повышения степени усвоения изучаемого материала модифицирована традиционная схема проведения лабораторных работ. Каждое занятие, включающее в себя, как обычно, объяснение преподавателем темы и целей работы, освоение пользователями технических и программных средств, используемых в работе, выполнение самостоятельных заданий, заканчивается выдачей преподавателем индивидуальных (бригадных) задач. Разработанные пользователями программы отлаживаются на следующем занятии, это повышает динамичность лабораторного курса. В соответствии с целями и задачами курса и учетом описанных принципов проведения занятий лабораторные работы охватывают следующий круг тем.

Тема 1. Знакомство с техническими и программными ресурсами ЭВМ.

Основные понятия: текущий диск; директорий; файловая система; отладчик программ; резидентные и транзитные команды ОС.

Основные этапы: загрузка системы, холодный и теплый старты; редактирование командной строки; изучение директив командного языка; простейшие операции над файлами (создание, переименование, удаление и др.); инструкции отладчика; изучение структуры директории.

Задание. Выполнить средствами командного языка несколько операций над файлами; объяснить логику выполнения используемых директив и изменения в директории диска.

Тема 2. Использование системных устройств ЭВМ.

Основные понятия: системная функция; ввод-вывод символа; буферизуемый ввод-вывод; состояние консоли; трассировка программы.

Основные этапы: изучение функций ввода-вывода символов и символьных строк; изучение инструкций пошаговой отладки; отладка учебных фрагментов программ с использованием изученных системных функций.

Задание. Разработать программу с использованием нескольких системных функций ввода-вывода.

Тема 3. Работа с файловой системой ОС в прикладной программе.

Основные понятия: сектор; группа; экзент; код завершения операции; адрес ПДП.

Основные этапы: изучение файловых функций ОС; разработка и отладка фрагментов программ, выполняющих различные операции над файлами.

Задание. Разработать программу, выполняющую ввод имени файла с клавиатуры микроЭВМ, поиск в директории, чтение-запись файла и др.

Тема 4. Разработка простой системы управления файлами.

Выполнение этой итоговой работы предполагает реализацию каждым пользователем (бригадой) логически законченного фрагмента с последующей комплексной отладкой основной программы.

Основываясь на результатах проведения цикла лабораторных работ, можно утверждать, что для проведения одной лабораторной работы целесообразно выделить четыре академических часа. Опыт проведения занятий показал, что, несмотря на небольшой объем практикума, пользователи, выполнившие работы по всем темам, уверенно ориентируются в вопросах организации и функционирования ОС микроЭВМ и получают практические навыки их использования, достаточные для самостоятельного изучения и освоения других ОС.

Заключение

Материалы настоящей работы демонстрируют возможность организации изучения и получения практических навыков использования ОС при относительно небольших затратах на реализацию. Экономичность предлагаемого подхода определяется не только выбором для изучения ОС CP/M, но и ориентацией на сетевую организацию класса. Эффективность рассматриваемой в работе локальной сети достигнута за счет широкого применения дополнительных уровней файловых подсистем. Существенной особенностью разработки является комплексный подход к решению задачи обеспечения **надежной работы** учебного класса: использование резидентно хранимого ядра ОС CP/M, внешних сменных ПЗУ и энергонезависимых ОЗУ квазидисков, автоматическое определение режима работы класса.

Целесообразность принципов, заложенных в организацию класса и методику проведения занятий, была подтверждена в процессе проведения лабораторного практикума. Несмотря на то, что описанная система разрабатывалась для учебных целей на кафедре МО ЭВМ ЛЭТИ, ее можно эффективно использовать для решения достаточно серьезных прикладных задач.

Телефон, 234-89-26, Ленинград

1. Хаузер Д., Хирт Дж., Хоцкинс Б. Операционная система MS DOS. Популярное руководство: Пер. с англ. и дополн. А. Б. Пандре; Предисл. В. М. Брябина.— М.: Финансы и статистика, 1987.— 168 с.
2. Грушвицкий Р. И., Коровацкий В. П., Преображенский А. В. Особенности построения и использования учебных микроЭВМ. // Микропроцессорные средства и системы.— 1985.— № 1.— С. 76.
3. CP/NET Control Program for a Microcomputer Network. User's Guide.— Digital research.— USA, 1980.— 186 p.

Статья поступила 4.02.88

УДК 681.327

М. Н. Егорличенко, Ф. Ф. Каримова, А. А. Барабанщикова

ВОСЬМИКАНАЛЬНЫЙ КОНТРОЛЛЕР ДЛЯ ПОДКЛЮЧЕНИЯ ТЕРМИНАЛОВ

В настоящее время выпускается большой набор устройств для подключения терминалов с интерфейсом ИРПС к интерфейсу «Общая шина» (СМ8514, СМ8502, СМ8529.03, СМ1420/602), однако при включении дополнительных терминалов к комплексу возникают значительные трудности. Для решения задачи был разработан и реализован восьмиканальный контроллер, соединяющий терминал с интерфейсом ИРПС и УВК СМ1420.

При его разработке ставились следующие задачи: обеспечение максимально простого подключения к комплексу;

использование минимальных затрат оборудования при хорошей надежности устройства.

Контроллер выполнен на двух печатных платах типа Е2 второй очереди СМ ЭВМ, поэтому его можно подключать в свободные места блока БМ1 процессора СМ2420 или в блок расширения системы типа СМ1420.0111 (межплатные соединения необходимо выполнить через разъем Х3).

В операционной системе контроллера восемь отдельных устройств типа СМ1420/602 (имя YL), поэтому стандартное ПО не надо дорабатывать.

В контроллере можно выделить следующие узлы: дешифрации адреса и кода операции на «Общей шине», синхронизации, программирования универсальных приемопередатчиков, анализа состояния каналов ИРПС, захвата «Общей шины» для прерывания, восемь каналов ИРПС.

В контроллере предусмотрена возможность программировать режим работы универсальных приемопередатчиков. Поступление сигнала BUS INITL на «Общую шину» устанавливает их в сброшенное состояние; снятие вызывает начало цикла программирования (на входы данных всех приемопередатчиков подается слово режима, а затем слово команды). При этом задается режим приема управляющих слов.

Контроллер запрограммирован на асинхронную приемопередачу с делением частоты 1:16 восьмьбитовыми посылками, дополненными до нечетного числа, с одним стоповым битом (слово режима 01011110, слово команды 0001011). Скорость передачи определяется переключкой (9600, 4800, 2400, 1200 и 600 Бод).

Терминалы подключаются УВК СМ1420 через контроллер и в составе комплекса имеют базовые адреса от 760000 до 777770, устанавливаемые переключкой или прожигом микросхемы КР556РТ4. Каждый терминал для обмена информацией использует четыре программно-доступных регистра, адреса которых определяются базовым

адресом терминала и смещением адреса (последняя триада).

Контроллер проверяется в составе УВК СМ1420 на текстах терминалов. При задании режимов работы текстов его следует считать стандартным контроллером СМ1420/402.

Использование общедоступной и недефицитной элементной базы позволяет легко воспроизвести данный контроллер при подключении терминалов с интерфейсом ИРПС к комплексу СМ1420.

625022, Тюмень, Солнечный проезд, 21; тел. 9-52-82

Статья поступила 2.02.88

УДК 681.3.06.519.68

В. В. Борисенко, Д. В. Варсанюфьев, А. Г. Дымченко

Ф-ПРАКТИКУМ

Ф-практикум — инструментальная среда для программирования на Фортране, разработанная в лаборатории Вычислительных методов механико-математического факультета МГУ под руководством А. Г. Кушниренко.

Стиль взаимодействия человека с Ф-практикумом опирается на идеи объектно-ориентированного (в терминологии [1]) подхода и интерактивного синтеза программ [2]. Из традиционной последовательности «редактирование—компиляция—сборка—выполнение» в Ф-практикуме остались только редактирование и выполнение. Компиляция программы производится автоматически и незаметно для человека одновременно с редактированием текста. Заметим, что в получивших в последнее время распространение турбосистемах [3] стадии компиляции и сборки лишь ускорены, но не устранены.

В качестве языка в Ф-практикуме использовано подмножество стандартного Фортрана-77 [4] (из существенных ограничений отметим отсутствие типов COMPLEX, DOUBLE PRECISION и оператора EQUIVALENCE). Добавлен стандартный де-факто оператор INCLUDE.

Редактирование

Работа в Ф-практикуме похожа на работу в Е-практикуме [5]. Предусмотрены возможности вставки структурных единиц Фортрана (PROGRAM, SUBROUTINE, FUNCTION, DO, IF—THEN—ELSE—ENDIF и т. п.) с помощью нажатия на специальную клавишу. В отличие от Е-практикума вставленные таким образом конструкции (кроме строки END) не защищены от текстового редактирования. Разумеется, внутри секции управляющие конструкции можно вводить и в текстовом виде (по буквам).

Обнаруженные в процессе компиляции ошибки немедленно выводятся на поля в правой части экрана:

PROGRAM TEST	!
CALL ПРОГР(X)	!ОШ. ТИП ПАРАМ
END	!
SUBROUTINE ПРОГР(I)	!
END	!

Если исправить ошибку (например, описав X как целую или I как вещественную переменную), то сообщение об ошибке автоматически исчезнет.

Выполнение

Программу без ошибок (с «чистыми полями») можно выполнить. При этом экран делится на две части. В верхней части изображается выполняемый фрагмент программы. Нижняя часть экрана занята окном для

операторов ввода-вывода. При выполнении операторов присваивания результат визуализируется на полях:

```
PROGRAM TEST
TYPE *, ' ВВЕДИТЕ M,N
ACCEPT *,M,N
TYPE *, 'НОД( ' ,M, ' , ' ,N, ' )=' ,NOD(M,N)
END
FUNCTION NOD(I,J)
M = I
N = J
I CONTINUE
IF(M.EQ.N) GOTO 2
IF(M.GT.N) THEN
M = M-N

```

ВВЕДИТЕ M,N
333,74
НОД(333, 74)= 37

Как обычно [5], программа выполняется по шагам или целиком. Кроме того, можно выполнять программу так же, как в традиционной Фортран-системе (весь экран отведен для ввода-вывода, а текст программы не виден). Ф-практикум диагностирует попытки использования переменных до того, как им присвоены значения, и выход адреса за границы массивов (контроль адреса, а не индекса позволяет работать с частями массивов, как с отдельными массивами в традиционном фортрановском стиле).

Заключение

Созданная в Ф-практикуме программа при выходе сохраняется в обычном текстовом файле. Она компилируется обычным Фортран-компилятором и выполняется средствами ОС. Таким образом, Ф-практикум применим и для разработки программ, которые будут выполняться вне его.

УДК 681.326.7

Г. Н. Соловьев, Б. Н. Ковригин, В. М. Сидуков, В. Г. Тышкевич, Р. К. Мифтахов, М. А. Иванов

УНИВЕРСАЛЬНАЯ ЛАБОРАТОРНАЯ УСТАНОВКА ДЛЯ ОБУЧЕНИЯ ПРОЕКТИРОВАНИЮ УЗЛОВ И УСТРОЙСТВ ЭВМ

Установка предназначена для макетирования логической структуры цифровых устройств. В ее состав входят следующие ячейки:

органов управления, предназначенные для включения питания, задания управляющих последовательностей, наблюдения сигналов и индикации состояния исследуемых схем;

цифровых элементов, библиотека которых в основном воспроизводит функциональный состав серий K155, K531, K555 и K599;

функционально-логических блоков на основе секционированных микропроцессорных БИС серии K1804;

сигнатурного анализатора для тестового и функционального диагностирования.

Дополнительно необходимы осциллограф и источник питания 5 В.

Входы и выходы логических элементов ячеек подключены к коммутационным гнездам с помощью соединительных шнуров. Ячейки составляют первый уровень конструктивных модулей установки. Второй уровень представлен четырьмя панелями размерами 237×485 мм с вертикальными перегородками, разделяющими их плоскости на вертикальные ряды, в которые устанавливаются ячейки различных размеров.

Габаритные размеры установки 1270×800×700 мм, масса 45 кг, стоимость 8000 руб. Конструктивно ячейка состоит из несущей платы с разъемом, коммутационными гнездами и

При редактировании-компиляции понятие «время ответа» отсутствует (диагностика и рекомпиляция измененного фрагмента производится мгновенно). Скорость интерпретации внутреннего представления программы в 10...20 раз ниже, чем в оптимизирующем Фортране (и в 5...10 раз выше, чем в интерпретирующем БЕЙСИКе). Объем разделяемого кода на машинах серии PDP-11 около 40 Кбайт. При объеме оперативной памяти 56 Кбайт возможна разработка Фортран-программ размером около 500 строк. В более благоприятной ситуации (УК НЦ, СМ4, СМ1420, «Электроника 85», «Электроника 79») этот размер увеличивается до 1,5...2 тыс. строк.

Ф-практикум выполнен на языке Си (около 10 тыс. строк). Благодаря реализации непосредственного драйвера текста [6] Ф-практикум может одновременно работать на терминалах практически любых типов.

Телефон 939-17-86, Москва

ЛИТЕРАТУРА

1. Ершов А. П. Об объектно-ориентированном взаимодействии с ЭВМ // Микропроцессорные средства и системы.— 1985.— № 3.— С. 2.
2. Barstow D, ed. Interactive programming environments.— NY: McGraw Hill, 1976.
3. Turbo C. Owner's Manual.— Borland International, 1987.
4. Катцан Г. Язык Фортран 77.— М.: Мир, 1982.
5. Варсановьев Д. В., Кушниренко А. Г., Лебедев Г. В. Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники // Микропроцессорные средства и системы.— 1985.— № 3.— С. 27—32.
6. Варсановьев Д. В., Дымченко А. Г., Кушниренко А. Г., Лебедев Г. В. Непосредственный драйвер текста НДТ-83 и системы на его базе // Микропроцессорные средства и системы.— 1987.— № 1.— С. 39—43.

Статья поступила 12.05.88

КРАТКОЕ СООБЩЕНИЕ

светодиодами для индикации состояния последовательных элементов ячейки, сменной печатной платы с микросхемами, вставляемой в разъем несущей платы, накладной фальшпанели с нанесенным на ней условным графическим обозначением схемы.

Таким образом, легко расширяется или полностью изменяется состав библиотеки цифровых элементов без изменения основной конструкции, что увеличивает срок службы установки, предоставляет пользователю большие возможности по совершенствованию функционально-логических и схемотехнических решений, развитию лабораторного практикума.

Установка используется на кафедре ЭВМ МИФИ для проведения практикумов по курсам Схемотехника и процессоры ЭВМ; Микропроцессорные устройства и системы; Надежность, контроль и диагностика вычислительной техники.

115409, Москва, Каширское шоссе, 31, МИФИ, каф. ЭВМ; тел. 324-34-35

Сообщение поступило 25.04.88

КОНСУЛЬТАЦИОННО-ТЕХНИЧЕСКИЙ ЦЕНТР

Консультационно-технический центр (КТЦ) по применению микропроцессоров (МП) предлагает потребителям МП БИС следующие услуги.

1. Информационные: представление материалов информационно-справочного фонда по применению изделий микроэлектроники (БИС МП, ПЗУ, РПЗУ, ОЗУ, АЦП, ЦАП и др.). В фонд включены руководства по применению и технические описания свыше 700 типов БИС.

Познакомиться с информацией можно следующим образом:

в читальном зале КТЦ (услуга оказывается бесплатно); пересылкой почтой или выдачей на руки материалов фонда (платная услуга, стоимость одного издания 15..50 руб.);

на договорной основе (цена договорная, в зависимости от объема поставляемой информации). С указателем фонда можно ознакомиться в КТЦ или запросить его по почте (рассылается бесплатно).

КТЦ располагает базой данных «Номенклатура МП БИС, рекомендованных для применения в аппаратуре народного хозяйственного назначения».

2. Ускоренное обучение основам микропроцессорной техники и ПЭВМ.

Проводится с помощью учебных и технических видеофильмов: «Микропроцессоры. Вводный курс» (10 телевизионных лекций); «Основы ПЭВМ. Средства отладки микропроцессоров» (11 телевизионных лекций); комплекта технических видеофильмов (всего 29 видеофильмов по автоматизации оборудования, машин, приборов).

Используются две формы обучения: на базе КТЦ (без предоставления жилья); самостоятельное обучение (КТЦ поставляет заинтересованным предприятиям на видеокассетах в стандарте VHS учебные программы, оказывает помощь в организации обучения на технических средствах заказчика и проводит необходимые консультации. Видеопрограммы просматриваются на видеомагнитофоне «Электроника ВМ-12» или любом импортном аналогичного типа).

Стоимость обучения или поставки видеопрограмм установлена прейскурантом, который можно получить в КТЦ бесплатно.

3. Консультации по применению изделий микроэлектроники (БИС МП, однокристалльных ЭВМ, РПЗУ, ПЗУ, ОЗУ, ЦАП, АЦП и др.).

Направления консультаций: общего профиля (выбор оптимальной номенклатуры БИС для систем автоматизации, коммерческие вопросы применения изделий микроэлектроники, изготовители и поставщики и др.);

по схемотехническим особенностям применения БИС, методам и средствам отладки МП-устройств;

по программному обеспечению МП-систем.

Консультации проводятся в помещении КТЦ с использованием технических средств БЕСПЛАТНО.

4. Технические услуги.

4.1. Поставка потребителям библиотек прикладных стандартных программных модулей для МП типов: КР580, КР1816ВЕ48/35, КР1816ВЕ51/31, КМ1821ВМ85. В зависимости от комплекта поставки цена за один пакет 190..2500 руб.

4.2. Разработка аванпроектов систем автоматизации машин, приборов, оборудования на базе МП-техники (проборботка архитектуры системы управления, выбор оптимальных технических средств, проработка технических и технико-экономических требований).

4.2.1 Разработка аппаратного и программного обеспечения МП-систем и ПЭВМ по техническому заданию заказчика.

4.3. Поиски исполнителя под задачу заказчика, занимающегося автоматизацией оборудования и приборов.

4.4. Организация и проведение конференций и семинаров по применению МП-техники.

Полный перечень услуг КТЦ Вы можете получить по почте или при посещении КТЦ.

Наш адрес: 107497, Москва, Щелковское шоссе, 79, Головной консультационно-технический центр по применению микропроцессоров; тел. 468-13-70, 468-81-75, 468-55-52

УДК 681.3.06

И. А. Гитин

ПРОГРАММА ФОРМАТИРОВАНИЯ ДИСКОВ С ПОВРЕЖДЕННОЙ ПОВЕРХНОСТЬЮ

Если Вы работаете на микроЭВМ СМ1800 и применяете ДОС1800, то для форматирования гибких дисков Вам, конечно, приходилось использовать процедуру FORMAT. Обычно она не доставляет хлопот. Но если диск немного поврежден, то процесс форматирования прерывается (во всяком случае, при работе с ДОС1800 версии 1.0). То же самое происходит при использовании программы DCOPY. Хорошо, если удалось отформатировать достаточное число дорожек и можно работать, изобретая различные трюки, чтобы «обойти» эту неприятность.

Для решения проблемы создана программа, работающая как в режиме проверки качества поверхности диска без разрушения записанной информации, так и в режиме форматирования (или реформатирования).

Качество диска проверяется непосредственно после инициализации; затем на нем создаются системные файлы ДОС1800. Если диск содержит абсолютно непригодные участки поверхности (секторы), то программа формирует объединенный файл, позволяющий исключить обращения к сбойным секторам при дальнейшей эксплуатации диска. Файл сбойных секторов создается с атрибутами невидимости и защиты от записи.

Программа различает абсолютно непригодные секторы и секторы, для обращения к которым понадобилось несколько попыток, т. е. «обреченные». Пользователь вправе причислить последние к абсолютно непригодным и включить в состав файла сбойных секторов. Предусмотрена возможность отказа от формирования файла сбойных секторов.

При форматировании диска программа изменяет фактор смещения по сравнению со стандартным для ДОС1800, что повышает скорость обращения к диску. Опыт эксплуатации программы показал, что нежелательные побочные эффекты при этом полностью отсутствуют.

320600, Днепропетровск, 5, ул. Симферопольская, 21, УГППКИ «Металлургавтоматика», отдел № 6, Гитину И. А.; тел. 47-46-55

Сообщение поступило 24.02.89

УДК 681 325—181.48

СТАТИЧЕСКОЕ ОЗУ K564PY2

Интегральная микросхема K564PY2 представляет собой КМОП ОЗУ с про-

извольной выборкой на 256 однобитных слов. Имеет прямой и инвертированный выходы, схему управления. Для выбора одного слова необходимо восемь входных адресных шин. Данные считываются без разрешения.

Предусмотрены выводы для выполнения команд считывания-записи цифровой информации и разрешения выбора микросхемы. Условное графическое обозначение ОЗУ дано на рис. 1, назначение выводов показано в табл. 1, электрическая структурная схема приведена на рис. 2.

Таблица истинности ОЗУ K564PY2

Режим работы	Вход			Выход	
	15	16	12	13	14
Запрет выбора	X	1	X	Z	Z
Считывание	0	0	X	Q	Q
Запись Лог. 0	1	0	0	Z	Z
Запись Лог. 1	1	0	1	Z	Z

Примечание. X — произвольное логическое состояние; Q, Q — прямая и инвертированная информация; Z — состояние высокого выходного импеданса.

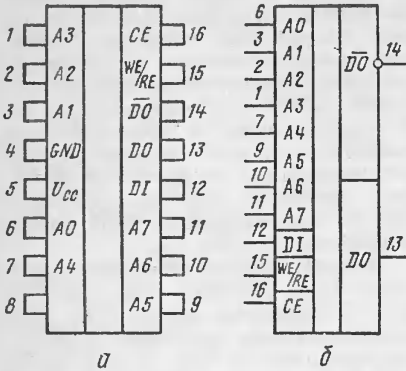


Рис. 1. Условное графическое изображение ОЗУ K564PY2 по порядку расположения (а) и функциональному назначению (б) выводов

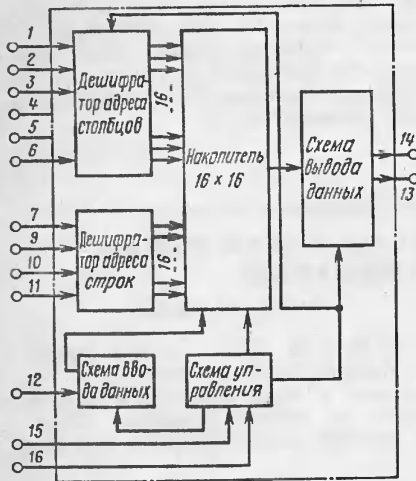


Рис. 2. Структурная схема ОЗУ

Таблица 1

Назначение выводов

Вывод	Назначение
1...3	Адресные входы столбца А3...А1
4	Общий вывод GND
5	Напряжение питания U _{CC}
6	Адресный вход столбца А0
7	Адресный вход строки А4
8	Свободный
9...11	Адресные входы строки А5...А7
12	Информационный вход DI
13	Информационный выход неинвертированный DO
14	Информационный выход инвертированный DO
15	Вход сигнала записи-считывания WE/RE
16	Вход сигнала разрешения CE

Таблица 3
Электрические характеристики ОЗУ K564PY2

Наименование параметра	Обозначение	Норма				Погрешность, %	Напряжение питания, В
		K564PY2A		K564PY2B			
		не менее	не более	не менее	не более		
Потребляемый ток в режиме хранения, мкА	I _{DD5}		10 5		200 200	±5	10 5
Ток утечки Лог. 1 на входах, мкА	I _{L1H}		0,05 0,05		0,05 0,05	±20	10 5
Ток утечки Лог. 0 на входах, мкА	I _{L1L}		0,05 0,05		0,05 0,05	±20	10 5
Ток утечки на информационных выходах, мкА	I _{LDO}				1 1,5	±5	10
Ток Лог. 0 сигнала выходной информации, мА	I _{DOL}	2,7 1,6		2,7 1,6			10 4,5
Ток Лог. 1 сигнала выходной информации, мА	I _{DOH}	0,9 0,9		0,9 0,9		±5	10 4,5
Емкость, пФ входная выходная	C ₁ C ₀		8 16		10 18	±30	10

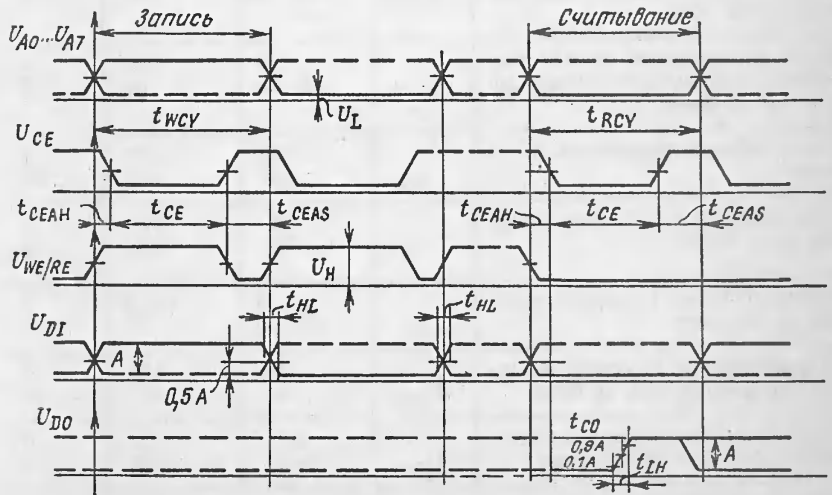


Рис. 3. Временные соотношения входных сигналов в режиме считывания-записи: все временные интервалы даны на уровне 0,5

Описание допустимых режимов эксплуатации ОЗУ К564РУ2

Наименование параметров режима	Обозначение	Норма			
		К564РУ2А		К564РУ2Б	
		не менее	не более	не менее	не более
Напряжение источника питания, В	U_{CC}	3	15	4,5	15
Напряжение на входе, В	U_I	-0,2	$\pm 0,2$	-0,2	+0,2
Напряжение на выводе питания при включении и выключении, В	U_{CC}	-0,5	15	-0,5	15
Максимальные токи Лог. 0 и Лог. 1 сигнала выходной информации, мА	I_{DOL} I_{DOH}	—	10	—	10
Максимальная допустимая мощность потребления в динамическом режиме при $T = +85^\circ\text{C}$, мВт	F_{DPP}	—	150	—	150
Длительность фронтов сигнала разрешения, мкс	t_{CELN} t_{CEHL}	—	1,0	—	1,0
Максимальная емкость нагрузки при $U_{CC} = 10\text{ В}$ на частоте 1 МГц, пФ	C_L	—	500	—	500

Таблица 5

Временные соотношения параметров и основные характеристики электрических режимов ОЗУ К564РУ2

Параметр	Обозначение	Норма при $T = 25 \pm 10^\circ\text{C}$		Напряжение источника питания, В
		К564РУ2А	К564РУ2Б	
Время цикла, нс, не менее записи	t_{WCY}	1500	2100	5
		650	1100	10
считывания	t_{RCY}	1500	2100	5
		650	1100	10
Длительность сигнала разрешения, нс, не менее	t_{CE}	1200	1500	5
		450	800	10
Время восстановления сигнала разрешения относительно сигнала адреса, нс, не более	t_{CEAS}	200	500	5
		180...200	200	10
Время установления сигнала разрешения относительно сигнала адреса, нс, не более	t_{CEAH}	1 100	100	5
		0...20	100	10
Время выборки разрешения, нс, не более	t_{CO}	1200	1500	5
		450	800	10
Напряжение Лог. 0 входных сигналов, В, не более	U_L	0,5	0,5	5
		0,6	0,6	10
Напряжение Лог. 1 входных сигналов, В, не менее	U_H	4,4...5	4,4...5	5
		9,3...10	9,3...10	10
Напряжение Лог. 0 сигнала выходной информации, мВ, не более	U_{DOL}	10	10	5
		10	10	10
Напряжение Лог. 1 сигнала выходной информации, В, не менее	U_{DOH}	9,99	9,99	10
		4,99	4,99	5
Напряжение источника питания, В	U_{CC}	3...15	4,5...15	

Все входы снабжены буферными схемами, минимизирующими их шунтирующее действие и обеспечивающими возможность непосредственного соединения ОЗУ с ТТЛ-схемами или другими логическими устройствами на МОП-схемах. Раздельные числовые шины входа и выхода устраняют взаимное влияние циклов считывания и записи. Выходы ОЗУ имеют три состояния, каждое из которых определяется командами считывания-записи и разрешения выбора микросхемы (табл. 2). В режимах записи и запрета выбора микросхемы числовые выходные шины отсоединяются от нагрузок, что позволяет связывать выходы нескольких ОЗУ по схеме проводного ИЛИ. Масса микросхемы не более 2 г.

Основные электрические характеристики ОЗУ К564РУ2 при $T = 25 \pm 10^\circ\text{C}$ приведены в табл. 3, предельные значения допустимых режимов эксплуатации — в табл. 4, временные диаграммы режима считывания-записи — на рис. 3, временные соотношения — в табл. 5.

Вход СЕ обеспечивает разрешение выбора конкретного ОЗУ из ряда микросхем в большой системе. Невыбранная микросхема остается в состоянии покоя, потребляя минимальную мощность. Информационный выход DO соответствует высокому уровню напряжения при хранении Лог. 1 и низкому — при хранении Лог. 0.

Г. Глушкова

КОМУ НУЖЕН РЕЖИМ ОЖИДАНИЯ?

Отклик на статью

В № 1 за 1988 г. нашего журнала была опубликована статья Е. А. Коноплева «Режим ожидания в накопителях на гибком магнитном диске», в которой рассматривались причины

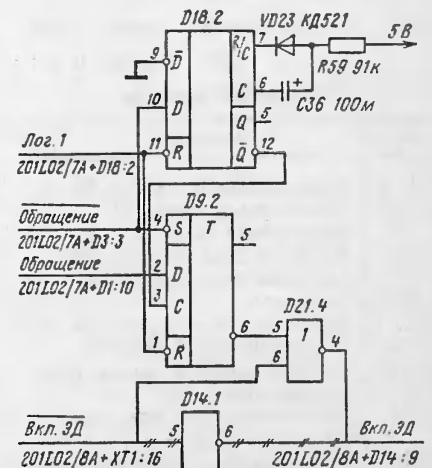


Рис. 1

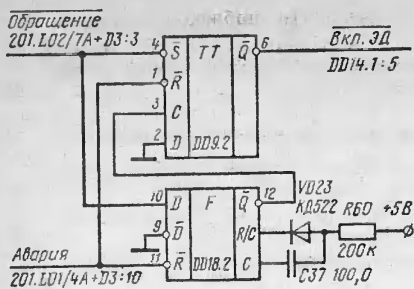


Рис. 2

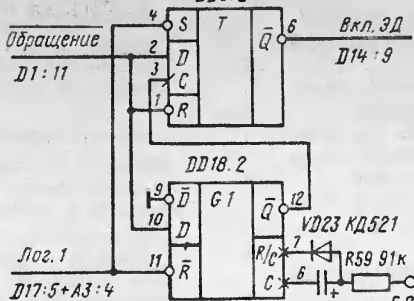


Рис. 3

напрасно износа гибких магнитных дисков (ГМД) и возможные пути его снижения. Такому износу подвергаются дискеты в накопителях, входящие в состав всех серийных моделей ДВК, комплектуемых контроллерами накопителей КНГМД. Один из недостатков контроллера — невозможность обеспечения режима ожидания, поэтому гибкий магнитный диск крутится в конверте, пока включена ПЭВМ.

Автором проведен анализ износостойкости дискет различных типов в реаль-

ных условиях работы: 8—10 ч в сутки и круглосуточно. Выяснилась парадоксальная ситуация: при гарантированном времени жизни дискеты BASF 2D 20 млн. проходов магнитной головки по одной дорожке (1140 ч, 70 лет службы) она живет в ДВК максимум три месяца!

После несложной доработки по приведенной в статье схеме срок службы дискет возрастает в 4—8 раз, достигая наилучших значений в случае совместной работы с драйвером НГМД МХ.

SYS М. И. Потемкина (№ 3 за 1988 г.). Применение только этого драйвера без аппаратной доработки накопителей не позволяет работать с дисководом индивидуально (при обращении к одному из них запускается весь пакет — до четырех накопителей, подключенных к системе).

На статью пришло несколько откликов. Отмечая необходимость введения режима ожидания, В. И. Кишиневский из Ульяновска, В. Я. Стародумов из Москвы, В. А. Тишкин из Таганрога предлагают упрощенные схемы доработки накопителей (рис. 1—3 соответственно). Однако ни один из авторов не продвинулся дальше первоначальной схемы, лишь упростили ее, хотя КНГМД используется с несколькими типами накопителей, кроме «Электроника 6021» (6022): Изот 5088, Изот 5321 (М), Роботрон 5640, «Электроника 6121», «Электроника 5305», где также необходимо обеспечить режим ожидания. Ни в одном из них нет свободных элементов, которые использовал Е. А. Коноплев при доработке НГМД «Электроника 6021».

УКАЗАТЕЛЬ СТАТЕЙ, ОПУБЛИКОВАННЫХ В ЖУРНАЛЕ ЗА 1989 ГОД

Общие вопросы развития информатики

- Гейман Л. М. — Ранние этапы развития информатики как системы знаний 3
 Гиглавы А. В., Немец А. В. — Освоение технологии САПР-БИС в КНР 3
Форум «МП»
 Иванов С. М., Шахнов В. А. — Достижения НРБ в области микропроцессорной техники и программного обеспечения 2
 Лишак Е. В. — Байтоторговля: прошлое, настоящее, будущее 2
 Карась И. З. — Работы по созданию законодательства информатики в СССР 1

МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

- Антонюк Б. В., Омельчук О. В., Присяжнюк В. Н., Терещенко Г. Г. — Тестовый процессор для мультипроцессорных систем 5
 Артюхов В. Г., Кондратюк В. А., Макеенок А. Н., Пенской В. И., Стацюра Е. В. — Цифровой процессор KM1813 BE1: Комплекс средств автоматизации проектирования 1
 Бобков В. А., Кушерев В. Н. и др. — Контроллер ЦАП 6
 Бондаренко А. А., Скороходов В. Ф. — Асинхронный контроллер динамического ОЗУ 1
 Бруевич Д. А. и др. — Одноплатное ОЗУ 128Кбайт с контролем хранения данных 6
 Галуев Г. А. — Специализированный матричный процессор для обработки бинарных изображений 4
 Кобылинский А. В., Калатинцев В. М., Заика А. И. — Микропроцессор KM1810VM87 с плавающей запятой 5
 Кузнецов Д. Н. и др. — КРОНОС: семейство процессоров для языков высокого уровня 6
 Лангуев В. В., Болотов С. А., Ольшак А. И. — Комплект БИС серии KM1823 для микропроцессорных систем управления двигателями внутреннего сгорания 2

- Мутанов В. И., Гришина Ю. П., Чернявский Ю. Г., Чернышев В. И. — Процессор цифровой обработки сигналов с векторной системой команд 4
 Розенштейн Э. П., Овчинников А. В. — Синхронизация работы микропроцессоров серии K1801 5
 Секушин Е. В., Осипенко В. И., Копыл П. А. — Универсальный программируемый контроллер прерываний 1
 Сидоренко В. П., Яровой С. И., Хоружий А. А. — БИС РПЗУ с УФ-стиранием информации серии K573 4
 Стянов А. И., Хорошунов В. С., Ачкасов В. Н. — Двухкристальный микропроцессор серии K1831 3
 Шмат В. К., Усов Н. Н. — БИС форматора данных для управления СБИС ЗУ ЦМД 2
 Шмат В. К., Кокорев Д. В., Усов Н. Н. — Контроллеры внешних запоминающих устройств на ЦМД 2

ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

- Бармин А. В., Каганов М. И., Кондрашев А. В. — ДВК как персональная АПЛ-машина 4
 Горшков Д. В., Зеленко Г. В., Шишкин А. В. — Микро 16 — одноплатная ПЭВМ на основе микропроцессора KR810VBI86 4
 Кравчук В. Г., Некрасов А. А., Флорентьев В. В. — Опыт работы с профессиональными персональными компьютерами ЕС1840 4
 Лосев И. С. — Будущее компьютерных игр 1
 Медведев В. В. — Аналоговый интерфейс ПЭВМ АГАТ 4
 Пинчук Н. И., Тищенко В. Д., Шалугин С. С., Школяренко А. К. — Программно-аппаратный комплекс для подключения печатающих устройств типа ЕС7040 к ПЭВМ ЕС1840, ЕС1841 4

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

- Абрамов С. М., Пименов С. П., Абакумов А. А., Хаткевич М. И. — Компьютерный вирус 2

Авдеев В. Н., Белов А. Е., Жедь А. Ю., Корнюшко В. Ф. — Интерпретатор языка БЕЙСИК со встроенным графическим пакетом для ДВК	5	Степанов В. Н. — Простейший мультизадачный планировщик	6
Алифанов Б. Ю., Дубравин В. Ю. — Программная система генерации текста заявки на изобретение	4	Христов П. В. — Архитектура эскорт-машины и ее аппаратная поддержка	3
Байдан И. Е., Гильян Н. Г. А., Гаджиев М. М., Депрес Д. С. Г. — Комплекс программного обеспечения для схемотехнических разработок на микроЭВМ	6	Инструментальные средства для отладки программ	
Барковский Д. В., Жарков А. П., Креславский И. Г. — Программный пакет МЕТАКС — инструмент для генерации ассемблеров	4	Айтъян С. Х., Гаурян К. П. — Инструментальные средства разработки экспертных систем на ПЭВМ	6
Белицкий Р. И. — Ядро операционной системы мультипроцессора с магистральной структурой	4	Балюк В. В., Кобзар С. П. — Диагностирование устройства на базе микропроцессора К580ИК80А	6
Бельчинский И. Л. — Перенос ядра ИНМОС между машинами различных архитектур	3	Белов В. А. — Построение экспертной системы средствами DDBASE11	6
Богатырев В. А. — Арбитр с децентрализованным кодовым управлением	5	Белоусова Г. В., Болмазов И. В., Кулаков В. Б., Несветайлов В. В., Федорова Н. А. — Инструментальная система отладки программ	1
Бурдин В. Р., Карасев В. И., Коломыц В. Г., Макагон С. С., Чернявский А. Д. — Распределенная обработка данных с использованием СУБД КВАНТ	5	Дмитриев В. А., Бедарев А. В., Гравов В. С., Бурнашев А. Д., Сапо С. Н. — Стенд для изучения временных соотношений сигналов	1
Вуколиков В. М., Дроздов Ю. Н., Колобков В. В., Лазарев А. В., Панкрац Е. П. — Пакет программ управления приборами через магистраль МЭК 625.1 для ПЭВМ	1	Иванцов С. Я. и др. — Автоматизированная контролирующая система ПАКС — МК	6
Гришин В. А. — Работа с датами в базе данных комплекса РТК МИКРО	4	Кацман В. Е., Коновая Я. М. — Система подготовки программ для персональных ЭВМ	6
Дианов А. П., Щелкунов Н. Н. — Средства и методы программирования	6	Любимов В. Г., Меламед О. Н. и др. — Отладочный комплекс для однокристалльных микроЭВМ семейства «Электроника С5»	6
Жарков Е. А., Кольцов В. И. — Кросс-система программирования 8- и 16-разрядных микропроцессоров, использующая унифицированный формат объектных модулей	6	Мамджян Г. Г., Наумчик М. П., Прохоренко А. Н., Розман Б. Я., Горбов Д. С., Рузанова Н. И. — Комплекс микропроцессорных модульных средств	1
Захаренков С. Я., Шехтер И. К. и др. — Программное обеспечение базовой АКИС для аналоговых и аналого-цифровых интегральных микросхем	6	Мельник В. Е. — ДВК для отладки и контроля цифровых устройств	3
Казменко С. В. — База данных при интеграции диалоговых средств конечного пользователя	5	Мусагалиев Б. Г. — Пульт отладки	6
Казменко С. В. — Несколько этюдов разработки задач в среде электронных таблиц	6	Пашенков С. Б., Стерин М. Ф. — Разработка программ для МП К580ИК80 с использованием кросс-системы «Электроника 85»	6
Каштанов Е. Ю., Корчак А. Е. — Подсистема печати документации «ПРИН» для ПЭВМ	6	Саркисов И. П. — Система автоматизированной разработки алгоритмов и программ для микроЭВМ «Искра 226» (система АРАП)	4
Корытный И. М., Мухалов И. О. — Диалог, основанный на принципе непосредственного редактирования информации при доступе к реляционной базе данных	5	Сафир М. Д., Сиянко В. М., Урсатий М. К. — Система отладки для секционированных микропроцессоров	6
Креславский И. Г., Барковский Д. В., Жарков А. П., Сидницын Н. В. — Микропрограммный ассемблер МАСС	4	Серебрянников О. Ю., Шевченко Е. Г., Леонидов-Каневский Е. В. — Автоматизированная система диагностики технического и программного обеспечения микропроцессорных систем	3
Крюков В. А. — Анализ принципов объектно-ориентированного программирования	2	Локальные сети и средства межмашинной связи	
Лилен А. Л. — Комплексные умножения на микропроцессорах К1815Ф3	6	Гайдашенко Н. И. — Резервированный интерфейс для устройств связи с объектом	6
Липаев В. В. — Некоторые особенности подготовки специалистов по проектированию программных средств	1	Дьяченко А. М., Королев В. М., Симаненков С. И., Цейтлин Б. В. — Станция локальной сети «Электроника МСВ3301»	6
Мастерова Г. А., Поляквца Т. А. — Программное обеспечение для организации диалога	6	Ермыкин В. И. — Устройство ввода информации в микропроцессорную систему	5
Музалевский И. В. и др. — Построение арбитров на ПЗУ	6	Каяченко М. А., Малахов В. В., Федупов А. С., Борисов В. В. — Устройство обмена массивами данных между интерфейсами КАМАК и ОШ	6
Ненашев М. А., Петрыкин Ю. С. — Системы управления базами данных для персональных ЭВМ	3	Князев А. И., Шестимеров С. М. — Электронный диск с прямым доступом к памяти ДВК2М	6
Никлаус Вирт — От разработки языков программирования к конструированию компьютеров	4	Лучук Д. А. — Синхронно-асинхронный адаптер с выходом на стык С2 для ЭВМ типа «Электроника 60»	6
Новиков В. В., Орехов А. В. — Программное обеспечение интерактивной трассировки печатных плат на ДВКЗМ2	6	Новиков Ю. В. — Универсальный параллельный интерфейс для модульных микропроцессорных систем измерения, контроля и управления	6
Пашенков С. В., Стерин М. Ф. — Мобильная файловая система с совместимым с UNIX программным интерфейсом	6	Стацук И. П., Александрович А. П., Захаревич А. В. — Аппаратно-программный интерфейс для обмена информацией между ПЭВМ ЕС1840 и ДВКЗ	6
Полянский П. В. — Организация прикладных баз данных на ПЭВМ с малой емкостью ОЗУ	5	ЯП Паскаль — вопросы реализации	
Рак В. В., Родылюк А. М. — Универсальная подпрограмма байтового деления	4	Добриневский С. Ф. — Процедуры обработки прерывания в Паскале ОС ДВК	5
Семеренко В. П., Гуменюк Я. А. — Программа расчета сигнатур на ПЭВМ	6	Грибов И. В., Зиновьев С. В., Шумаков А. В. — Программирование систем реального времени на языке Паскаль	5
		Горшенин А. Г., Воронов В. Б. — Звуковое сопровождение программ в Паскале М86 для ПЭВМ ЕС1840	5

Базы данных

Акулов В. К., Журавлев В. И., Новожилов Н. А., Шинкевич С. Л.— Система управления реляционными базами данных 2
 Волков Д. К.— МикроСУБД на языке БЕЙСИК для ПЭВМ 2
 Максимьяк С. П., Сорокин Ю. Ю., Субач В. В.— Непосредственный доступ к данным в файлах на Паскале-1 2

Периферийные устройства микроЭВМ

Алешин А. Н., Крюков С. Н.— Контроллер КНМЛ для микроЭВМ «Электроника 60» 1
 Андрушин О. С.— Сопряжение ДВК с устройством ввода графической информации типа СМП6410 5
 Анашкин И. И., Гагарин В. В.— Многофункциональный информационно-управляющий комплекс на базе одноплатных ЭВМ 1
 Асанов Р. Ш., Никитин А. С.— Имитатор интерфейса МПИ 6
 Бескровный В. Г., Урсатьев А. А., Сапожникова С. Л., Шестаковский М. Л.— Накопитель на кассетной магнитной ленте СМ5211—внешнее ЗУ микроЭВМ «Электроника 60» 6
 Белильников В. Д., Визин Ю. И.— Контроллер пульта оператора микропроцессорного измерительного прибора 6
 Вознюк Л. А., Горожин А. Д.— Программная эмуляция дисплея ВТА 2000 на базе дисплея «Символ» 6
 Гайдучок Р. М., Шумский Г. А.— Символьно-графический дисплей на базе газоразрядной индикаторной панели 1
 Гринь Н. Ф., Уткина Т. Г.— Модем связи с бытовым магнитофоном 5
 Грязнов Д. О., Омельяненко А. Х., Шевчук Ю. В.— Электронный диск в вычислительном комплексе под управлением ОС, совместимой с ОС UNIX 1
 Давыдов В. А.— Одноплатный контроллер магнитной ленты 6
 Канцеров В. А., Родионов В. В., Стригин В. Б., Халько Н. Н.— Устройство сопряжения дисководов ЕС5074 с контроллером дисководов PL X 45D 1
 Корнев В. А., Пройдаков Э. М., Щелкунов Н. Н., Дианов А. П.— Универсальный контроллер на базе ОЭВМ КР1816ВЕ51 1
 Котельников Ю. Н., Крочев А. М., Куликов А. С., Никитин С. П., Обходов Е. А.— Технологические контроллеры серии УТК6 1
 Кузнецов А. Ф.— Устройство ввода графической информации в ЭВМ 4
 Мруга А. Д., Щербачев Д. Ю., Пинчук В. М.— Простой контроллер алфавитно-цифровой клавиатуры Накалов Е. Ф., Тюлькин С. П.— Универсальный интерфейс ввода-вывода с элементами самопроверки 6
 Новиков А. В., Климов В. П., Смирнов В. Н., Ковальков В. К.— Устройство последовательного интерфейса мультитеплексного канала 1
 Осипов Е. Н.— Совместная работа БИС КР156ХЛ2 и К1816ВЕ48 в системах управления бытовой радиоэлектронной аппаратурой 4
 Падиряков Ю. А.— Текстовый экраный редактор SED для дисплея «Электроника МС 7401» 1
 Хохлов Д. А., Масленков М. Я., Крюков М. Г.— Использование БИС КР580ГФ24 в задающем генераторе дисплейного модуля 1
 Андреев С. В., Ходулев А. Б.— Вывод графической информации на точечно-матричные печатающие устройства 3
 Антонов Ю. А., Бибило П. Н., Гольдберг Е. И., Каркоцкая И. П., Чигирь Н. П.— Система логического проектирования цифровых устройств на программируемых матричных БИС 3
 Красильников И. В., Приходько П. С., Чурков В. М., Щетинин Ю. И.— Микросхемы программируемой матричной логики серии КР1556 2

Чурков В. М., Захарова Л. Е.— Подсистема проектирования ПЛМ 2
 Чурков В. М., Котрелев С. А., Захарова Л. Е.— Программное обеспечение САПР цифровых устройств для программируемой матричной логики серии КР1556 2
 Чурков В. М., Котрелев С. А.— Подсистема проектирования на ЭПЛК 2
 Чурков В. М., Красильников И. В.— Подсистема проектирования на ПЛМ 2

Машинная графика

Алексеев Г. И., Мильников С. П.— Интерактивная графическая система ГРАФ 1
 Бобков В. А., Кислюк О. С., Боровина Т. А.— Диалоговая графическая система для исследования структур сложных молекулярных объектов МОЛЛЮСК-2 2
 Бучнев А. А., Сизых В. Г., Минин В. Ф.— Цветная полутонная станция профессионального назначения ГАММА — 7.1. 5
 Васюков С. Т., Коломеев О. А., Перекрест О. В., Ситников Г. Ф., Ткач С. Е.— Цветной монитор с высоким разрешением для растровых дисплеев 5
 Вельтмандер П. В., Власов С. И.— ОФОРИС — прикладная графическая система формирования иллюстративных материалов 1
 Вельтмандер П. В., Жуков Г. В., Килина Л. В. и др.— Системы автоматизированного конструирования для персонального АРМ 2
 Голиков К. П., Педанов И. Е., Шурий В. Н.— Подготовка инженерной документации с помощью системы РЕДГРАФ 2
 Данилов Н. Ю., Полукаров А. Д., Столяров М. В.— Дисплейные микроконтроллеры семейства МИДИКОН Коллегов М. В., Репков В. В.— Дисплей с высоким разрешением 5
 Лацис А. О., Штаркман В. С.— Системы команд растрового графического дисплея «Электроника МС7401» 5
 Маркова Н. А.— Графический стандарт и его внедрение на ПЭВМ 1

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Азаров А. Д.— Большая операционная система 4
 Бабанов И. И.— Малая дисковая операционная система 4
 Бобков Г. М.— Видеоконтроллер СМ18107006 4
 Бобков В. А., Лангуев В. В., Болотов С. А., Ольшак А. И., Божко Н. Н., Каневский В. Г., Мостов И. С., Трилесник Е. А.— Программное обеспечение межмашинного обмена микроЭВМ СМ1800 4
 Бубело Н. И.— Язык программирования для микроЭВМ 4
 Бубович С. С.— Устройство формирования синхрипульсов и символов в изображении для ТВ-систем с микроконтроллером 4
 Бушуев С. Д., Диктерук М. Г., Жунусов Д. Ж., Иносов С. В.— Локальный регулятор температуры на основе ОЭВМ серии К1816 4
 Васильев Н. И., Пройдаков Э. М., Диделева Н. А., Левитина Г. В. и др.— Особенности реализации ОС МИКРОС-86 4
 Водовозов В. М., Горячев Ю. А., Пожидаев А. К.— Комплексная система ЧПУ на базе ДВК 2
 Вяткин С. А.— Тестовая операционная система 4
 Гадалов А. Ф., Борзов Г. В., Ильин Е. П.— Система редактирования, передачи управляющих программ и контроля для станков с УЧПУ «Электроника НЦ-31» 2
 Гальченко А. А., Самойлов В. В.— Результаты измерения производительности ЭВМ 4
 Горбачев В. С., Липпинг В. Э.— Система трехмерной машинной графики для моделирования промышленных роботов 2

Горбачев В. С., Полуобояринов О. Д.— Использование учебной микроЭВМ УМПК-80М в лабораторном практикуме по роботехнике	3	интерфейсом КТС ЛИУС-2	3
Горбачев В. С., Полуобояринов О. Д., Зуев М. В., Гительсон А. В.— MINI—VAL: язык автономного программирования учебного робота «РОБКО-01»	3	Орестов Ю. А., Бобылев В. Н.— Программируемый микроконтроллер на основе ОЭВМ КР1816ВЕ31	5
Глухов В. И.— Модуль системного контроля микроЭВМ СМ1810	4	Падирыков Ю. А., Белинский В. Т., Журило В. А.— Модуль системы обработки данных реального времени	5
Гуськов В. Д., Еремеев А. Л., Чукалов П. Б.— Центральный процессор микроЭВМ СМ1810	4	Погосян М. О., Агинян Г. С., Арутюнян В. Г.— Варианты построения генераторов прямоугольных импульсов с программируемой длительностью	4
Данильченко Н. Т., Лозовой Л. Н.— Перенастраиваемая микропроцессорная система управления самоходным рентгеновским дефектоскопом «Сирена-1»	1	Полянский П. В.— «Электроника БК-0010» в системах исследования объектов с распределенными параметрами	3
Джиган В. И.— Линия связи управляющей микроЭВМ с модулями антенной решетки	3	Резников Б. Г., Таратута А. У.— Переключатель интерфейса для микроЭВМ СМ1800	4
Духин В. В., Меньшикова Е. Г., Смирнов Н. В.— Программное обеспечение микропроцессорного контроллера комплекта группового вождения тракторов	3	Руцков М. В.— Мультипортовая память для построения видеокомплексов	2
Зайцев В. А., Кротов С. Н., Прошкин С. Д., Савченко А. В.— Мультимикропроцессорная система спектрального анализа сигналов в реальном масштабе времени	5	Сайкин В. Т., Артюх И. Г., Карпов И. И., Юданов В. А.— Учебный робот	2
Иванов А. П., Осьмаков А. Н., Лысенко В. В.— Система для сбора, обработки и регистрации информации на кассетной МЛ	5	Севостьянов А. В., Давыденко Ю. М.— Простой пультовый дисплей на микропроцессоре КР580ВВ79	6
Исаев А. Н., Блинные В. И., Кривошеев М. И., Сарьян В. К., Красносельский И. Н., Корытов В. В., Боловинцев Ю. М.— Передача патентной информации в составе телевизионного сигнала	3	Черняк С. И., Табаткин В. М.— Индикатор шин микропроцессора К1810ВМ86	6
Казанцев А. П., Майоров Л. Н., Данилов А. В.— Указатель информации и интерфейс цветного телевизора для микроЭВМ «Электроника БК-0010»	5	Трусов В. А., Гаврилюк С. Ю.— Контроллер микропроцессорной системы зажигания автомобильного двигателя «Электроника МС2713»	3
Казанцев А. П., Данилов А. В.— Интерфейс пользователя ПЭВМ «Электроника БК-0010»	5	Щелкунов Н. Н., Дианов А. П.— Организация однокристалльных микроконтроллеров	5
Коарига Д. С., Гладченко С. М.— Расширение программного обеспечения для промышленного робота «Электроника НЦТМ-01»	2		
Коломиец В. Г., Королев П. В.— Система автоматизированного кассового расчета столовой предприятия на базе ДБК	6		
Кольнер Г. Г., Маслов С. Г.— Кросс-система программирования для однокристалльных ЭВМ	4		
Кузьмичев С. В., Набутовский А. М., Селезнев В. П.— Система подготовки программ для программируемых контроллеров	2		
Лангуев В. В., Ольшак А. И., Гаврилюк С. Ю.— Процессор КМ1823ВУ1	3		
Лангуев В. В., Болотов С. А., Трусов В. А.— Устройство ввода-вывода КМ1823ВВ1	3		
Львов К. В., Гирба И. А.— Подключение к ЭВМ «Электроника ДЗ-28» устройств с последовательным интерфейсом ИРПС	3		
Макаров И. М. и др.— Учебная гибкая производственная система	2		
Матвеев В. Ф., Никитюк В. А., Прокопенко В. К., Шевцов П. И.— Программируемый контроллер на базе однокристалльной ЭВМ КМ1816ВЕ48 для группового управления станками с ЧПУ в ГПС	1		
Матвеев В. И., Староверов Ю. Г.— Бинарная система технического зрения для промышленного применения	2		
Монахов В. Т.— Программное обеспечение ПЭВМ «Электроника БК-0010»	3		
Мифтахов М. Г., Боргасевич В. С., Мифтахова Г. Р.— Многопроцессорная информационно-управляющая система с каналом ПДП для геофизических исследований скважин	5		
Овчаренко А. И., Пушко П. В., Хилобок В. И., Зверев В. И.— Микропроцессорная система контроля датчиков зажигания	3		
Овчаренко А. И.— Применение таймера КР580ВВ83 для измерения временных интервалов при исследовании датчиков зажигания	3		
Овчаренко А. И., Хилобок В. Е.— Преобразователь параметров датчиков-распределителей зажигания с			
		Медицинские приложения	
		Вайнштейн Л. Г.— Вычислительный комплекс для обработки электрофизиологической информации в условиях нейрохирургического стационара	5
		Захаров С. В., Марфенко К. С., Подлепецкий Б. И., Торубаров С. В.— Система на базе МПК БИС КР580 и КМ1813ВЕ1 для мониторинга физиологических параметров человека	5
		Лукьянов В. И., Логинов О. Е., Даньков В. Б., Тетерина Е. А., Бородин С. М., Блинков С. М., Никандров М. Г.— Информационно-измерительная система для исследования времени двигательных реакций человека	5
		УЧЕБНЫЙ ЦЕНТР	
		Борисенко В. В., Варсанюфьев Д. В., Дымченко А. Г.— Ф-практикум	6
		Грушвицкий Р. И., Преображенский А. В., Симанов А. В., Яккер В. М.— Класс учебных ЭВМ в лабораторном практикуме «Операционные системы»	6
		Дядькин А. И., Кушнир В. Е., Панфилов Д. И., Шаронин С. Г.— Способы построения и исследования интерфейсов устройств ввода и отображения информации микроконтроллеров на примере БИС КР580ВВ79	3
		Зырянов А. Г., Котегов С. С., Кудрявцев И. В. и др.— Скоротечение и ЭВМ	6
		Пройдаков Э. М.— Введение в мониторы микроЭВМ	3
		В блокнот разработчика	
		Сунгуров В. С., Ионов Н. Н.— Доработки БИС серии КР5888	5
		Фейзханов У. Ф.— Ошибка в операционной системе для ДВКЗ	5
		В СТРАНАХ — ЧЛЕНАХ СЭВ	
		Михальски А., Свидер В.— Семейство графических устройств ввода-вывода	6
		Синкевич Т.— Система отладки MSWP для 8-, 16- и 32-разрядных микропроцессоров	6
		Справочная информация	
		Интерфейсные БИС микропроцессорного комплекта К1801: микросхема К1801ВП1-035	1
		Рекомендации по применению микросхем серии К1801	2
		Микросхема РПЗУ К573РФЗ	3
		Оперативное запоминающее устройство К565РУ7	4
		Постоянное запоминающее устройство КР1801РЕ2	5
		Оперативное запоминающее устройство К564РУ2	6

УДК 621.3.049

Кузнецов Д. Н., Недоря А. Е., Тарасов Е. В., Филиппов В. Э. **Кронос: семейство процессоров для языков высокого уровня** // Микропроцессорные средства и системы.— 1989.— № 6.— С. 2.

Дан обзор архитектуры семейства процессоров Кронос и ОС для них. Цель разработки — создание универсального процессора с аппаратной поддержкой языков высокого уровня для конструирования ЭВМ открытой архитектуры: от встроенных микроЭВМ и однопроцессорных рабочих станций до многопроцессорных ЭВМ класса супер-мини.

УДК 681.326.35.77

Бобков В. А., Кушарев В. Н., Свиридович В. С., Чернуха Б. Н., Черныковский Д. Н. **Контроллер цифро-аналогового преобразователя KP588BT5** / Микропроцессорные средства и системы.— 1989.— № 6.— С. 4.

Рассмотрены состав, технические характеристики и основные режимы работы контроллера ЦАП, входящего в состав микропроцессорного комплекта БИС серии KP588.

УДК 681.323

Михальски А., Свицер В. **Семейство графических устройств ввода-вывода**. // Микропроцессорные средства и системы.— 1989.— № 6.— С. 10.

Институтом систем управления в г. Катовице разработано семейство графопостроителей и устройств преобразования графической информации различных форматов. В данной статье описывается графопостроитель MDG-1, который предлагается на советском рынке. Даются технические данные, рассматриваются схемные решения и управляющая программа, хранящаяся в ПЗУ. Графопостроитель MDG-1 предназначен для работы в качестве графического устройства выхода для персональных компьютеров в медицине, школах, научных исследованиях.

УДК 681.3.066

Мастерова Г. А., Полякова Т. А. **Программное обеспечение для организации диалога и документирования в системах автоматизации на основе микроЭВМ «Электроника 60»** // Микропроцессорные средства и системы.— 1989.— № 6.— С. 24.

Рассмотрены принципы организации ввода исходных данных и документирования результатов без использования традиционного способа программирования с помощью операторов ввода-вывода.

УДК 681.3:657.47

Каштанов Е. Ю., Корчак А. Е. **Подсистема печати документации ПРИН для персональных компьютеров** // Микропроцессорные средства и системы.— 1989.— № 6.— С. 31.

Описывается подсистема печати документации, расширяющая возможности текстовых процессоров WordStar, REFOR, ТЕКСТ, Документ-2 при печати документов со сложной структурой рубрикации, с автоматическим созданием содержания, списка рисунков и таблиц, предметного указателя.

УДК 681.3:621.3

Новиков В. В., Орехов А. В. **Программное обеспечение интерактивной трассировки печатных плат на ДВКЗМ2** // Микропроцессорные средства и системы.— 1989.— № 6.— С. 32.

Разработан графический редактор и вспомогательные программы для интерактивной трассировки двухслойных печатных плат на микроЭВМ ДВКЗМ2. Вспомогательные программы готовят перфоленты для фотоплоттера и сверльного станка с ЧПУ, прорисовывают топологию платы на матричном АЦПУ, CM6325.

UDC 621.3.049

Kuznetsov D. N., Nedorya A. E., Tarasov E. V., Filippov V. E. **CHRONOS: the high-level language processor family**. // Microprocessor devices and systems.— 1989.— N. 6.— P. 2.

The survey covers the architecture of the CHRONOS processor family and operating system for them. CHRONOS processors were designed to become the universal data processing unit with hardware support of high-level language programming for open architecture computers: from the built-in microcontrollers and single-processor workstations to multiprocessor configurations of super-mini class.

UDC 681.326.35.77

Bobkov V. A., Kusharev V. N., Sviridovich V. S., Chernukha B. N., Chernyakovskiy D. N. **Digital-to-analogue converter controller type KP588BT5**. // Microprocessor devices and systems.— 1989.— N. 6.— P. 4.

The architecture, technical specifications and main operation modes of DAC controller LSI of the KP588 family are described.

UDC 681.323

Michalsky A., Swider V. **Graphic I/O devices family: Plotter MDG-1**. // Microprocessor devices and systems.— 1989.— N. 6.— P. 10.

The Polish Institute for Control systems (Katowice) have designed a family of plotters and digitizers for various worksheet size. This article features MDG-1 plotter which is available in the USSR. Its technical specifications, circuit and ROM-resident software is described. MDG-1 is designed to be used as graphic output device for personal computers in medicine, education and science.

UDC 681.3.066

Masterova G. A., Polyakova T. A. **The software units for dialogue and printout support for "Elektronika-60"-based automation systems**. // Microprocessor devices and systems.— 1989.— N. 6.— P. 24.

The principles of data input and printout organization without usage of traditional I/O operators is explained.

UDC 681.3:657.47

Kashtanov E. Yu., Korchak A. E. **Document printout utility PRIN for personal computers**. // Microprocessor devices and systems.— 1989.— N. 6.— P. 31.

The document printout utility PRIN increases power of WordStar, REFOR, TEXT and DOCUMENT-2 text processors for documents with complex rubrication structure. It automatically generates table of contents, subject index, and tables and illustrations list.

UDC 681.3:621.3

Novikov V. V., Orekhov A. V. **Interactive printed circuit routing system running on DBK3M2 microcomputer**. // Microprocessor devices and systems.— 1989.— N. 6.— P. 32.

The graphic editor and utility programs for double-sided printed circuit boards routing running on LSI-ii-like computer are developed. The utilities produce control programs on punched tape for photoplotter and drill machine and draw PCB layout on dot-matrix printer type CM6325.

УДК 681.3.069

Жарков Е. А., Ковальцов В. И., Максимов Г. М. Кросс-система программирования 8- и 16-разрядных микропроцессоров с унифицированным форматом объектных модулей // Микропроцессорные средства и системы.— 1989.— № 6.— С. 34.

Рассматривается реализация CUFOM-процессоров для мини-ЭВМ CM4. Описываются элементы кросс-системы программирования микропроцессоров КР1816, К580, К1810, ориентированные на универсальный формат объектных модулей CUFOM (кроссассемблеры, редактор связей, библиотека объектных модулей, настраивающий загрузчик). Подчеркивается совместимость кросс-систем программирования микропроцессоров для различного класса ЭВМ (CM4, DEC-10). Предлагаются дополнительные возможности, введенные в компилятор с языка BCPL.

УДК 681.3.06

Пашенков С. В., Стерин М. Ф. Разработка программ для микропроцессора К580ИК80 с использованием кросс-системы I85 // Микропроцессорные средства и системы.— 1989.— № 6.— С. 36.

Рассмотрен опыт эксплуатации кросс-системы I85 на примере разработки программ управления файловой системой на устройстве прямого доступа. Сделан вывод об эффективности использования кросс-системы, сокращении сроков разработки программ и повышения их качества. Особое внимание уделено совместимости с ИНМОС.

УДК 681.326.7

Иванцов С. Я., Клим В. Я., Мамай В. И., Посупонько Н. В., Терпугов Ю. В. Автоматизированная контролирующая система ПАКС—МК // Микропроцессорные средства и системы.— 1989.— № 6.— С. 40.

Описана многофункциональная малогабаритная система контроля цифровых узлов, предназначенная для проверки функционирования в реальном масштабе времени изделий, содержащих ТТЛ-, ТТЛШ- и КМОП-элементы, в том числе и СБИС микропроцессоров. Система удобна в эксплуатации, легко адаптируется к проверяемым изделиям. Имеются средства для диагностики неисправных объектов контроля.

УДК 681.325

Шелкунов Н. И., Дианов А. П. Средства и методы программирования ОМК // Микропроцессорные средства и системы.— 1989.— № 6.— С. 49.

Рассмотрена методика программирования и чтения ПЗУ, встроенного в ОМК семейства МС48 и МС51. Предложены программно-аппаратные средства программирования ОМК, совместимые с универсальным программатором мМС8102 и пакетом программного обеспечения МППС для него. Система программирования ориентирована на работу в ОС ISIS (ДОС1800).

УДК 681.326

Черняк С. И., Табаткин В. М. Индикатор шин микропроцессора К1810ВМ86 // Микропроцессорные средства и системы.— 1989.— № 6.— С. 67.

Описаны режимы работы и принципиальная схема индикатора шин микропроцессора К1810ВМ86. Устройство предназначено для индикации шины адреса, данных и управляющих сигналов в процессе аппаратной настройки устройств.

UDC 681.3.069

Zharkov E. A., Kovaltsov V. I., Maksimov G. M. CUFOM — a cross system for 8-and 16-bit microprocessor program development using unified object module format. // Microprocessor devices and systems.— 1989.— N. 6.— P. 34.

The implementation of CUFOM processors on CM4 mini-computer is discussed. The elements of the cross system for 8080 and 8086 processors using single object module format (crossassemblers, link editor, librarian and relocating loader) are described. The compatibility of different implementations of the cross-system on PDP-11 and DECSystem-10 is pointed out. The special extra features, supported by the BCPL language compiler, are proposed.

UDC 681.3.06

Pashenkov S. V., Sterin M. F. Program development for KP580IK80 microprocessor using I85 cross-system. // Microprocessor devices and systems.— 1989.— N. 6.— P. 36.

The experience of I85 cross-system usage is illustrated on the sample program design for direct-access device file management unit. The efficiency of the cross-system application which results in higher program quality at less development time is shown. Special attention is paid to system compatibility with ИНМОС Unix-like operating system.

UDC 681.326.7

Ivantsov S. Ya., Klim V. Ya., Mamay V. I., Posupon'ko N. V., Terpugov Yu. V. Automated monitoring system PAKS-MK // Microprocessor devices and systems.— 1989.— N. 6.— P. 40.

A compact multifunction monitor system for digital devices is designed for status monitoring in real-time scale. The system can monitor units containing TTL, TTLS and CMOS components, including microprocessor LSIs. It is simple in operation and can be easily configured for particular device under test. Some means for troubleshooting of damaged devices are also provided.

UDC 681.325

Shelkunov N. I., Dianov A. P. Methods and tools for single chip microcontroller programming. // Microprocessor devices and systems.— 1989.— N. 6.— P. 49.

The methods of internal EPROM data read and write for Intel MCS48 and MCS51 microcontroller families are explained. The hardware and software tools for internal EPROM programming and verification are proposed, which are compatible with uMC8102 programmer and its software support. The software runs under ISIS-II (DOS1800) operating system.

UDC 681.326

Chernyak S. I., Tabatkin V. M. 8086 microprocessor bus display. // Microprocessor devices and systems.— 1989.— N. 6.— P. 67.

The circuit diagram and operation modes of bus display unit for 8086 microprocessor are shown. The unit is designed for address, data and control bus display during hardware debug process.

Заместитель главного редактора

С. М. ПЕЛЕНОВ

Номер подготовили: Е. И. Бабич, Г. Г. Глушкова, В. М. Ларионова. Корректор Е. М. Кучерявенко. Технический редактор Г. И. Колосова

Адрес редакции журнала:

103051, Москва, Малый Сухаревский пер., д. 9А

Телефоны: 208-73-23, 208-19-94

Сдано в набор 22.08.89.

Подписано к печати 13.10.89.

Формат 84×108¹/₁₆ Т-16352

Офсетная печать

Усл. печ. л. 10,08. Уч.-изд. л. 14,6.

Тираж 108.850

Заказ 6226 Цена 1 р. 10 к.

Орган Государственного

комитета СССР

по вычислительной технике

и информатике

Набрано в ордена

Трудового Красного Знамени

Чеховском полиграфическом

комбинате Государственного

комитета СССР по печати 142300, г. Чехов Московской области

Отпечатано в Московской

типографии № 13

ПО «Периодика»

Государственного

комитета СССР по печати

107005, г. Москва, Денисовский пер.,

30. Заказ 323



ПАРАЛЛЕЛЬ
PARALLEL

СОВЕТСКО-
БРИТАНСКОЕ
СОВМЕСТНОЕ
ПРЕДПРИЯТИЕ

СП «ПАРАЛЛЕЛЬ»

— это гарантия
передовых технологий
в области автомати-
зации проектирования
во всех отраслях
народного хозяйства

Совместное советско-британское предприятие «Параллель» поможет Вам создать функционально законченные программно-технические комплексы на персональных ЭВМ для автоматизации проектирования в среде системы Auto CAD.

СП «Параллель» поставит Вам:

Auto CAD, версия 10 на русском или английском языке

Auto CAD AEC

Auto SHADE

Auto SOLID

ABASE

GLISP

SOCRAT

— система автоматизированного проектирования и выполнения чертежей (поставляется с Компилятором Автолиспа, в 5—20 раз повышающим быстродействие программ);

— дополнительный пакет программ, позволяющий использовать Auto CAD в соответствии со специфическими требованиями архитекторов;

— программа преобразования чертежей Auto CAD в реалистические цветные изображения с тенями (включает Auto FLIX — средство создания динамических изображений);

— программа пространственного твердотельного моделирования;

— комплекс программ, предназначенный для обработки в параметрическом виде графической, текстовой и табличной информации с целью информационной поддержки процесса проектирования;

— программное средство, предназначенное для автоматической генерации параметризованных программ на языке Автолисп;

— программное средство для конструирования прикладных экспертных систем.

СП «Параллель» — официальный дилер компании «Autodesk» в СССР. Продукция СП в СССР продается за рубли, за рубежом — за валюту. СП «Параллель» обеспечивает бесплатное сопровождение в течение двух лет с даты поставки собственных и фирменных программных продуктов, а также бесплатную замену программных продуктов новыми версиями.

В зависимости от количества приобретаемых копий, начиная с двух, СП «Параллель» устанавливает скидку от 20 % до 50 %.

Наш адрес: 107073, Москва, Стромьинский переулок, 23/7.
тел. 268-55-12. FAX — 230-26-01.

Кооператив «ЭЛИАС» и исследовательская группа программистов
ALEX SOFTWARE

ПРЕДЛАГАЮТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРОВ

- Оригинальную систему «SHIELD» для защиты программ от нелегального копирования
- Систему AGDS для создания игровых, обучающих и демонстрационных программ, мультфильмов
- Программу автоматизированного расчета заработной платы «Триоком-17»
- Тестовую программу для социально-психологических исследований «Портрет личности»
- Программу, позволяющую автоматизировать процесс проектирования микросборок, а также больших и малых интегральных схем в системе PCAD
- Программу, обеспечивающую обмен текстовой информацией между различными типами компьютеров серии СМ, ИЗОТ, WAX посредством магнитной ленты
- Информационно-справочную систему по обслуживанию автомашин
- Компактную систему «Beard», позволяющую получить справочную информацию (изображение и параметры) наиболее употребляемых элементов радиоэлектронной аппаратуры
- Программу «Корреспондент», помогающую автоматизировать канцелярские работы (регистрацию входящих и исходящих, ведение архивных документов, изготовление картотек)
- Программу «SP», предназначенную для графического изображения числовых последовательностей в режимах автоматического вычисления и произвольной установки масштабов
- Оригинальную систему «Ingertor» для защиты программ от вирусов
- Программу «filter», скрывающую Ваши файлы от постороннего прочтения

«ЭЛИАС» — ЭТО НАДЕЖНЫЙ И ВЫГОДНЫЙ ПАРТНЕР

НАШ АДРЕС: 127562 МОСКВА,
КАРГОПОЛЬСКАЯ УЛ.,
ДОМ 17. ТЕЛЕФОН: 903-04-57

