



МИКРО- ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

1 | 1988

ISSN 0233-4844

Одноплатная микроЭВМ на основе КМОП СБИС, а также БИС серий К588, К537, К1623РТ, совместимая по системе команд и интерфейсу с микроЭВМ «Электроника 60»

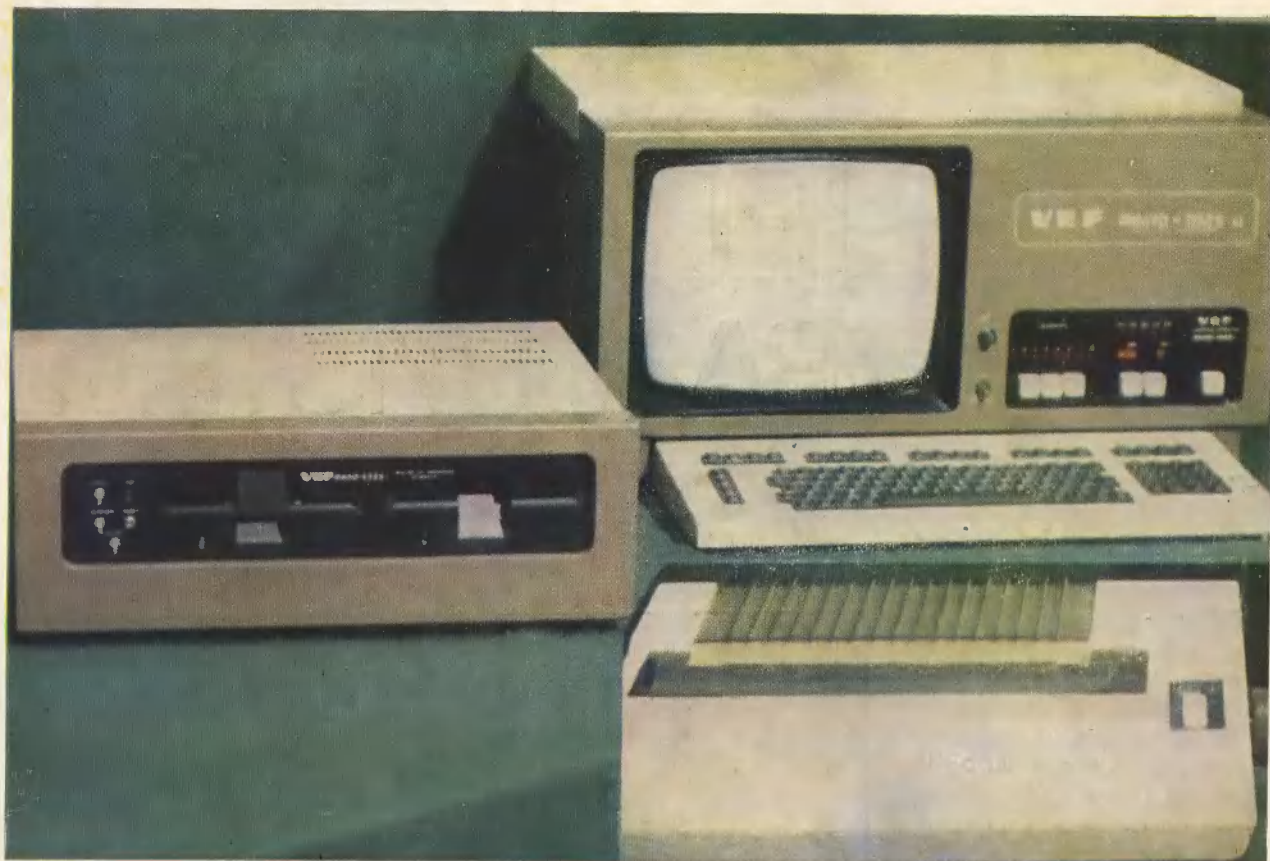
КМОП-комплект БИС серии К588: контроллер прямого доступа к памяти К588ВТ2, селектор адреса КР588ВТ1

Программируемый контроллер клавиатуры и индикации КР580ВВ79—устройство для построения систем отображения и ввода информации

Графические средства в четырех реализациях языка Бейсик для персональных ЭВМ «Искра 226», IBM PC, MSX, «Электроника 85»

Программная реализация международного графического стандарта ГКС для ПЭВМ





МИКРОЭВМ С МАРКОЙ «ВЭФ»

Универсальная микроЭВМ «ВЭФ-микро-1025» представляет собой управляющую ЭВМ для различных систем обработки и передачи информации, в том числе и производственных. Эта микроЭВМ с модульной архитектурой разработана на базе МПК БИС серии КР580. Модули ЦП, ОЗУ, ПЗУ, интерфейсов стандартных внешних устройств сопряжены унифицированной магистралью.

МикроЭВМ предназначена для использования в качестве базовой системы при разработке программ в процессе проектирования микропроцессорных комплексов, интеллектуальных терминалов, контроллеров, измерительных приборов на БИС серии КР580.

Программное обеспечение микроЭВМ — си-

стемный монитор, резидентный ассемблер, редакторы текста, трансляторы и интерпретаторы языков Бейсик, Кобол, Паскаль, дискровая операционная система, операционная система реального времени, программы генерации протоколов для передачи данных в МП-системах, библиотека программ пользователя.

Подробные аппаратные и программные характеристики микроЭВМ будут даны в статье В. А. Таланова «МикроЭВМ с маркой ВЭФ (опыт РПО ВЭФ)» в одном из очередных номеров журнала.

Телефон для запросов о технической документации и программном обеспечении — 27-60-44, Рига (коммерческий директор).

ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКЕ
И ИНФОРМАТИКЕ.

Издается с 1984 года

ММП МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД

НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ

1 / 1988 МОСКВА

СОДЕРЖАНИЕ МИКРОПРОЦЕССОРНАЯ ТЕХНИКА

ПЕРСОНАЛЬНЫЕ КОМПЬЮТЕРЫ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ Машинная графика

ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ

Измерительные системы и устройства

УЧЕБНЫЙ ЦЕНТР

Справочная информация

Ершов А. П. — Колонка редактора	2
Кобылинский А. В., Калатинец В. М., Заика А. И. — Программируемый контроллер клавиатуры и индикации КР580ВВ79	3
Бобков В. А., Горовой В. В., Лаврусенко В. А., Черняковский Д. Н. — Одноплата микроЭВМ на основе БИС серии КР588	8
Бобков В. А., Чернуха Б. Н., Черняковский Д. Н., Свиридович В. С., Кушарев В. Н. — Селектор адреса КР588ВТ1	11
Кушарев В. Н., Свиридович В. С., Чернуха Б. Н., Черняковский Д. Н., Бобков В. А. — Контроллер прямого доступа к памяти К588ВТ2	13
Иванов В. И., Клепов А. В., Костиков Д. В. — Малогабаритные персональные ЭВМ с жидкокристаллическим экраном — новый класс персональных ЭВМ	18
Кетков Ю. Л. — Графические средства в алгоритмическом языке Бейсик	25
Баяковский Ю. М., Михайлова Т. Н., Рыжова И. Г. — Реализация базовой графической системы для ПЭВМ	31
Кипаренко В. В. — Стандартизация интерфейсов микропроцессорных систем (Приложение)	35
Дианов А. П., Щелкунов Н. Н. — Модули программирования логических схем	40
Кочубин В. Б. — Двойная плотность записи на гибких магнитных дисках «Электроника ГМД-7012»	44
Коноплев Е. А. — Режим ожидания в накопителях на гибком магнитном диске	47
Кисилевский Я. Ф., Пикусов С. Г. — Использование двойной плотности записи на НГМД «Электроника ГМД-7012» при работе с микроЭВМ «Электроника МС 1201»	49
Мышкин И. Л., Щербаков А. С. — Интерфейс накопителя на гибких магнитных дисках «Электроника ГМД-7012»	51
Суранов А. Я., Госьков П. И., Якунин А. Г. — Микропроцессорная система измерения параметров бинарных изображений на основе многоэлементного фотоприемника	54
Крянев А. С., Белянкин П. В. — Устройство сбора аналоговой информации для микроЭВМ «Электроника 60»	56
Тимофеев Е. Ю. — Сопряжение персональных ЭВМ с приборным интерфейсом	58
Германенко А. В., Миньков Г. М. — Применение микроЭВМ «Искра 1256» для исследования быстропротекающих однократных процессов	62
Барметов Ю. П., Боев С. А., Евтеев Ю. И. — Анализатор сигналов акустической эмиссии на основе микропроцессорной системы	63
Торгов Ю. И. — Программируемый контроллер клавиатуры КР580ВВ79	65
Панфилов Д. И., Шаронин С. Г., Яковлев С. Е. — Сопряжение ЦАП и АЦП с микропроцессорными системами	72
Станкевич А. В. — Клавиатура и блок индикации для приборов на базе микропроцессора К1801ВМ1	76
Курмаев О. Ф., Балабанов А. А. — Контроллер динамической памяти	79
Петров Л. В., Ляшев В. П., Соловьева Т. А. — Устройство сопряжения накопителя на магнитных лентах с микроЭВМ «Электроника 60»	83
Джиган В. И., Лужецкий С. Г. — Модуль электрически перепрограммируемого ПЗУ для микроЭВМ семейства «Электроника 60»	85
Яблонский Ф. М., Усмонов У. Ю., Влахов С., Косев И., Павлов Д., Радонов Г. — Микропроцессорный комплекс для изучения средств отображения информации	89
Добуш Ю. Д., Старостенко О. В., Наптога Е. Е. — Клавиатура на базе однокристалльной микроЭВМ КМ1816ВЕ48	91
Давыдовская В. В., Козловский В. А., Малинин А. В., Усов Г. И. — Микросхема асинхронного статического ОЗУ КР537РУ13	93

Главный редактор
А. П. ЕРШОВ

Редакционная
коллегия

А. Г. Алексенко
В. М. Брябрин
А. А. Васенков
[зам. главного редактора]
И. В. Гельбицкий
А. Б. Венгероз
Г. Р. Громов
[ответственный секретарь]
В. П. Иванников
М. Б. Игнатев

А. В. Каляев
И. З. Карась
В. П. Куприянов
С. С. Лавров
В. В. Липаев
К. А. Меликян, И. А. Мизин
Б. Н. Наумов
[зам. главного редактора]
С. М. Пеленов
[зам. главного редактора]

А. К. Платонов
А. А. Попов
Д. А. Поспелов
Б. И. Рамеев
О. Л. Смирнов
А. А. Стогний
М. К. Сулим
Н. М. Шаруенко

Редакционный совет:

Р. Л. Ашастин
И. В. Бабынин
С. Н. Бушев
Е. П. Велихов
Н. Н. Говорун
В. В. Корчагин
В. П. Макаревич
И. И. Малашинин
Ю. Е. Нестерихин
А. Р. Назарян
А. Л. Нефедкин
И. В. Прангишвили
Л. И. Преснухин
В. В. Пржиялковский
Н. Л. Прохоров, Г. Г. Рябов
В. И. Хохлов
Н. Н. Шереметьевский
В. В. Шильдин, А. В. Яковлев
Э. А. Якубайтис

Номер подготовили:
Е. И. Бабич, Г. Г. Глушкова,
В. М. Ларионова, С. С. Матвеев
Фото О. В. Чиркина

Адрес редакции журнала:
103051, Москва,
Малый Сухаревский пер., д. 9А
Телефоны: 208-73-23, 208-19-94

Слано в набор 04.12.87. Т-05070
Подписано к печати 15.02.88
Формат 84x108¹/₁₆. Бумага № 1.
Высокая печать. Усл. печ. л. 10.08.
Уч.-изд. л. 14.8. Тираж 96 100 экз.
Заказ 371. Цена 1 р. 10 к.
Орган Государственного комитета
СССР по вычислительной технике
и информатике
Московская типография № 13
ПО «Периодика» ВО «Союзполи-
графпром» Госкомиздата СССР
107005, Москва, Денисовский пер.,
дом 30

На первой странице обложки—
МикроЭВМ «Электроника 85»

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАТИКА: ОТРАСЛЬ ИЛИ ИНФРАСТРУКТУРА?

Сколько я себя помню, по крайней мере, с середины 60-х годов специалисты по вычислительной технике и по еще не обретшей своего имени информатике горевали, что нет над нами подходящей государственной структуры. Нам бы члена Политбюро, отвечающего за вычислительную технику, свое министерство, отдельную строку в бюджете, а уж Курчатова и Королева мы бы выдвинули из своих рядов и принесли стране компьютерный паритет. Здесь была не только романтическая зависть к героической истории строительства ракетно-ядерного щита, были и естественное стремление к самоопределению и гражданская озабоченность замедлением темпов развития.

Настойчивость в постановке этого вопроса можно сравнить лишь с той определенностью, с которой это предложение проваливалось при попадании на высшие уровни принятия решений. Направивается естественное, на первый взгляд, мнение, что ушедшие в прошлое руководители отраслей сохраняли статус-кво, являясь либо носителями, либо орудиями периода застоя.

Перестройка и строительное осознание обществом жизненной важности развития промышленной информатики снова и снова ставят вопрос о ее организационно-экономической структуре. На ответственных совещаниях вновь говорится о том, чтобы собрать отрасль под одну крышу.

И все-таки надо подумать, не сохраняем ли мы верность принципам, которые сами изживают себя? Правильно ли бороться с ведомственностью в информатике путем создания еще одного ведомства?

Объективно отрасль информационной промышленности сложилась. В нее четко сбились объединяемые, например, ежегодными экономическими сводками журнала «Datamation» продукты микроэлектроники, микропроцессоры, ЭВМ всех моделей, периферия, аппаратура передачи данных, программное обеспечение, базы данных, сопутствующий сервис. Отрасль с высоким уровнем концентрации: 100 ведущих компаний стран рыночной экономики произвели в 1986 г. подавляющий объем товаров и услуг стоимостью порядка 200 млрд. дол.

Но вот что интересно! Переживая период становления и обособления, завоевывающая позиции, ресурсы и репутацию, поддерживая или, по крайней мере, учитывая общественное разделение труда, организации в то же время неуклонно диверсифицируют свою деятельность, двигаясь в целом в сторону применений информационных технологий.

С другой стороны, смежные информационные отрасли: связь, теле- и радиовещание, журнально-газетное и книгоиздательское дело, фото-, видео- и аудиоиндустрия все в большей степени смыкаются со всеми слоями электронно-вычислительно-программной промышленности. Не менее тесные связи прорастают со стороны таких промышленных гигантов, как машино- и приборостроение, прежде всего, благодаря встроеным компьютерам, интеллектуальным роботам, цифровым приборам и управляющим устройствам.

Вернемся еще раз к вдохновлявшим нас аналогиям. Современные атомная и космическая программы при всем их размахе и наукоемкости являются как бы «дважды» специализированными. Они создают машины, которые работают в очень специальных недоступных человеку средах и их «деловая» связь с обществом тоже весьма специальна: через сети электропередачи и через ракеты-носители. Это одновременно предписывает и облегчает обособление соответствующих промышленных отраслей. Положение информационной промышленности совсем иное. Она характерна прежде всего, огромной площадью соприкосновения буквально со всеми видами человеческой деятельности, а сам механизм этого соприкосновения предельно разнообразен и по форме, и по содержанию. Если продуктивно-ориентированные промышленные отрасли удовлетворяют потребности живого организма общества некоторым ассортиментом частных благ, то информационная промышленность дает этому организму саму нервную систему.

В этих условиях прямой перенос на эту глобальную и своеобразную отрасль рецептов и организационных форм, сложившихся в других условиях и для других целей, был бы, по меньшей мере, неосмотрительным. Призывая читателей к размышлению над будущим нашей отрасли, позволю себе высказать личное мнение: в любом случае информационная промышленность должна быть полицентричной. Отрасль категорически нуждается в множественности точек роста по целой совокупности причин. Одна из них — преодоление монополии производителя. Не победив ее, не поставив носителей идей и продуцентов в условия реального социалистического соревнования, конкуренции, если хотите, мы никогда не получим рублевых чипов памяти, безремонтных ПЭВМ, быстрых и бесшумных принтеров, вечных телефонов и неограниченно тиражируемого и удобного ПО.

А. Ершов

УДК 681.326.35.77

А. В. Кобылинский, В. М. Калатинец, А. И. Заика

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР КЛАВИАТУРЫ И ИНДИКАЦИИ КР580ВВ79

БИС КР580ВВ79 — программируемое устройство, предназначенное для построения систем (подсистем) отображения информации на основе дисплеев (индикаторов) различных типов, а также систем (подсистем) ввода информации из различного типа клавиатур (рис. 1, табл. 1).

Эта БИС позволяет автоматически сканировать клавиатуру, определить и выдать код позиции клавиши, нажать которой в матрице клавиатуры, подавить помехи дребезга контактов клавиатуры, а также регенерировать изображение на дисплее (индикаторе). Программирование восьми режимов ввода и четырех режимов вывода обеспечивает широкие возможности для пользователя при вводе и отображении информации. Вводимая информация может накапливаться в обратном магазине (ОМ) ОЗУ датчиков с доступом «первый вошел — пер-

вый вышел» емкостью 8 слов \times 8 разрядов (8 байт). При наличии информации в ОМ ОЗУ датчиков микросхема выработывает запрос прерывания, а в случае ввода или чтения более восьми символов — сигналы ошибок переполнения или переопустошения в слове состояния. Для регенерации изображения на дисплее (индикаторе) в микросхеме имеется ОЗУ отображения объемом 16 слов \times 8 разрядов. Время сканирования клавиатуры и регенерации изображения задается программным способом. По электрическим параметрам БИС КР580ВВ79 полностью совместима с микросхемами серии ТТЛ. Конструктивно выполнена в пластмассовом 40-выводном корпусе типа 2123.40—2 ГОСТ 17467—79.

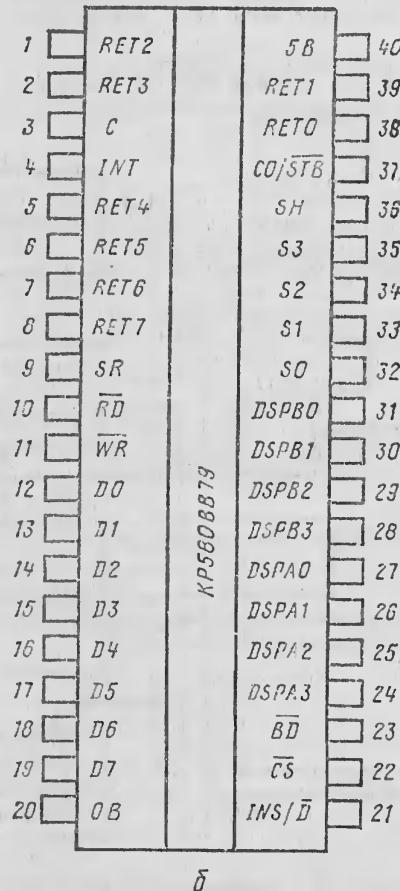
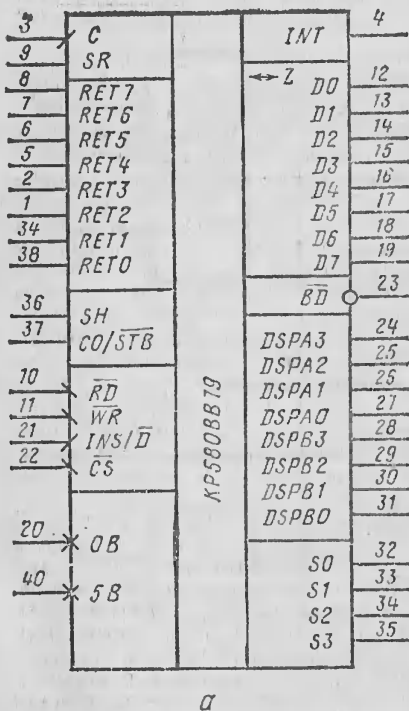


Рис. 1. Условное графическое обозначение (а) и назначение выводов (б) БИС КР580ВВ79

Основные электрические параметры БИС КР580ВВ79 в диапазоне температур $-10 \dots 70^\circ\text{C}$

Напряжение источника питания U_{CC} , В	$5 \pm 5\%$
Ток потребления I_{CC} , мА, не более	120
Входное напряжение, В: высокого уровня на линиях возврата U_{IH} , RET, не менее	2,2
высокого уровня на всех остальных входах U_{IH} , не менее	2,0
низкого уровня на линиях возврата U_{IL} , RET, не более	1,4
низкого уровня на всех остальных входах U_{IL} , не более	0,8
Выходное напряжение, В: высокого уровня на выходе «Запрос прерывания» U_{OH} , INT, не менее	3,5
высокого уровня на всех остальных выходах U_{OH} , не менее	2,4
низкого уровня U_{OL} , не менее	0,45
Выходной ток, мА: высокого уровня I_{OH} , не более	$-0,15$
низкого уровня I_{OL} , не более	1,9

Структура контроллера КР580ВВ79

БИС КР580ВВ79 организована на базе внутренней шины. В ее состав входят (рис. 2): ОМ ОЗУ датчиков со схемой управления и устранения дребезга клавиатуры, буферами клавиатуры и датчиков и схемой анализа состояния ОМ ОЗУ датчиков; схема управления и синхронизации со счетчиком сканирования; ОЗУ отображения с регистрами адреса и данных; средства сопряжения с системной шиной — буферы канала данных и схема управления чтением-записью.

ОМ ОЗУ датчиков обеспечивает хранение информации, поступающей со входов RET7... RET0, и выдачу ее в микропроцессор через внутреннюю шину данных и буферы канала данных по командам чтения. В этом блоке используются следующие сигналы:

RET7... RET0 — информация, вводимая с клавиатуры или матрицы датчиков. За счет внутренних нагрузок на этих выходах постоянно обеспечивается состояние «Лог. 1» до появления входных сигналов низкого уровня.

SH — входной сигнал, используемый в режиме сканирования клавиатуры для расширения (увеличения числа) функций клавиш (вывод SH также имеет внутреннюю нагрузку).

CO/STE — входной сигнал, назначение которого определяется режимом работы: в режиме сканирования (состояние «Лог. 1») увеличивает чис-

Таблица 1

Номер вывода	Назначение вывода	Обозначение	Тип вывода
1, 2	Линии возврата	RET2, RET3	Вход
3	Тактовый сигнал	C	Вход
4	Запрос прерывания	INT	Выход
5...8	Линии возврата	RET4...RET7	Вход
9	Сброс	SR	»
10	Чтение	RD	»
11	Запись	WR	»
12...19	Канал данных	D0...D7	Вход-выход с тремя состояниями
20	Общий вывод	OV	—
21	Команда / данные	INS/D	Вход
22	Выбор микросхемы	CS	»
23	Гашение отображения	BD	Выход
24...27	Канал дисплея А	DS PA3...DS PA0	»
28...31	Канал дисплея В	DS PB3...DS PB0	»
32...35	Линии сканирования	S0...S3	»
36	Сдвиг	SH	Вход
37	Управление / строб	CD/STB	»
38, 39	Линии возврата	RET0, RET1	»
40	Вывод питания от источника напряжения	U _{CC}	5 В

нирования клавиатуры при наличии данных в ОМ ОЗУ датчиков; в режиме сканирования матрицы датчиков при обнаружении изменения в датчиках.

Схема управления и синхронизации программно задает скорость ввода-вывода и синхронизирует работу всех узлов контроллера. Необходимая скорость ввода-вывода создается от деления частоты входной синхронизации на программно задаваемое число, лежащее в диапазоне 2...31. При работе схемы используются сигналы сброс SR, тактовый C и BD, блокирующий индикатор (дисплей) в момент смены кодов символов.

Счетчик сканирования — 4-разрядный двоичный счетчик сканирует строки клавиатурной матрицы и регенерирует изображение на дисплее (индикаторе). Программно счетчик можно установить в кодированный или декодированный (дешифрованный) режим работы. В кодированном режиме он выдает на линии сканирования S3...S0 4-разрядный двоичный код от 0000₂ до 1111₂, который дешифрируется внешними дешифраторами; в декодированном режиме он выдает два младших разряда кода, а на линии сканирования S3...S0 — 4-разрядный унитарный код (активный уровень — «Лог. 0»). Внешних дешифраторов в этом случае не

ло функций клавиатуры; в режиме стробирующего ввода (состояние «Лог. 0») используется как строб для ввода информации с линий возврата RET7...RET0 в ОМ ОЗУ датчиков

по фронту сигнала CO/STB (вывод CO/STB также имеет внутреннюю нагрузку).

INT — выходной сигнал переходит в состояние «Лог. 1»: в режиме ска-

Структурная схема БИС КР580ВВ79

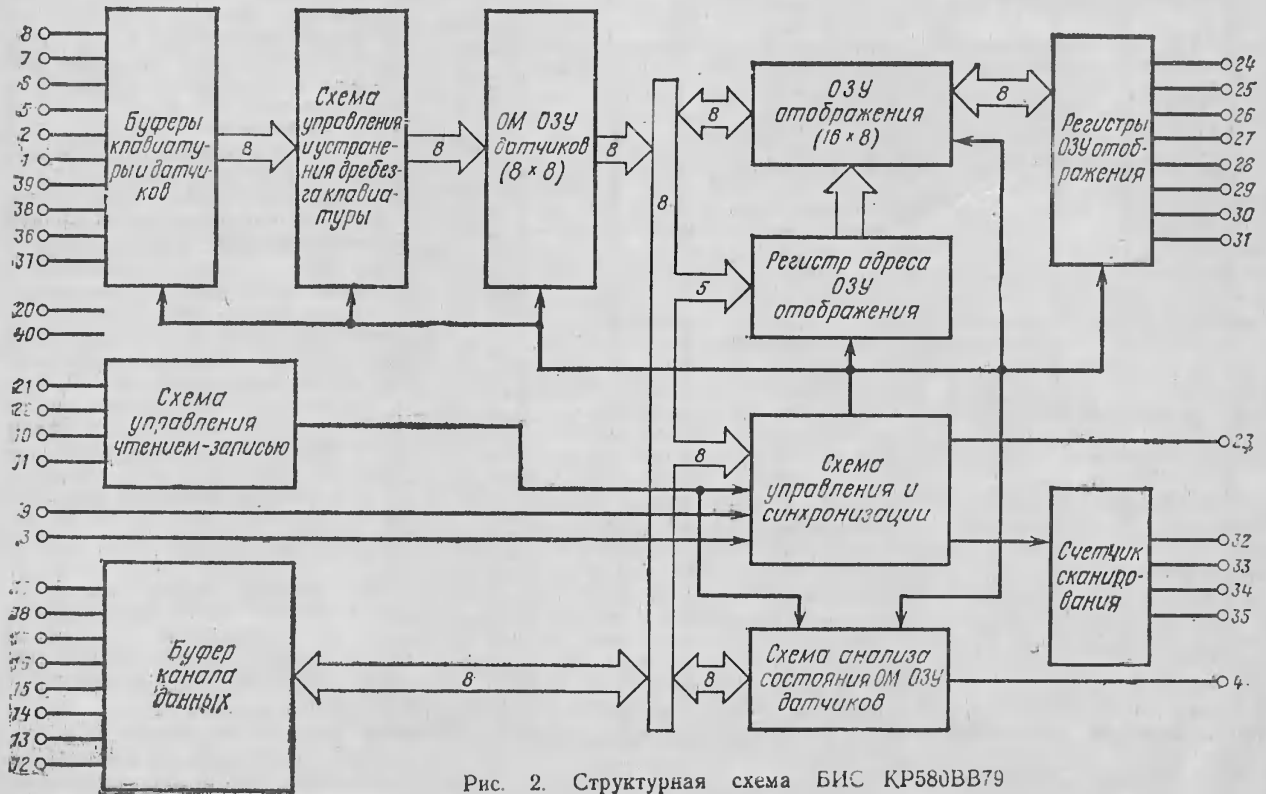


Рис. 2. Структурная схема БИС КР580ВВ79

Таблица 3

Команда	Код команды			Код операции				
	D7	D6	D5	D4	D3	D2	D1	D0
Режим работы	0	0	0	K4	K3	K2	K1	K0
Программирование скорости ввода-вывода	0	0	1	P4	P3	P2	P1	P0
Чтение ОМ ОЗУ датчиков	0	1	0	A4	X	A2	A1	A0
Чтение ОЗУ отображения	0	1	1	A4	A3	A2	A1	A0
Гашение — запрет записи отображения	1	0	1	X	Г3	Г2	Г1	Г0
Сброс	1	1	0	C4	C3	C2	C1	C0
Сброс INT — установка режима обнаружения ошибок	1	1	1	E	X	X	X	X

Примечание. X — безразличное состояние.

Связь с системной шиной

Операции чтения-записи выполняются в зависимости от состояния управляющих сигналов (табл. 2, рис. 3).

Программирование

Центральный процессор управляет контроллером с помощью команд

(табл. 3), которые передаются по системной шине данных при $\overline{CS}=0$ и $INS=1$ и загружаются по нарастающему фронту сигнала \overline{WR} . Команда «режим работы» устанавливает режим работы клавиатуры (табл. 4) и дисплея (табл. 5).

Команда «программирование скорости ввода-вывода» программирует

требуется, однако клавиатура может иметь не более четырех строк клавиш, а дисплей не более четырех знакомест при 8-разрядном коде символа и восемь знакомест при 4-разрядном коде.

Буферное ОЗУ отображения — ОЗУ с произвольным доступом хранит коды символов, отображаемых на дисплее. Каждый байт может рассматриваться как два независимых полубайта. Это позволяет обслужить дисплей с большими функциональными возможностями — на 16 знакомест при 8-разрядном коде символа и на 32 знакоместа при 4-разрядном коде символа. ОЗУ доступно для записи и чтения информации. Регистр адреса ОЗУ отображения может быть программно установлен в режим автоинкрементирования. В процессе регенерации отображения на дисплее информация из ОЗУ отображения циклически выдается на выходы DSPA3... DSPA0 и DSB3... DSB0, причем бит DSPA3 соответствует биту D7 шины данных, а бит DSPB0 — биту D0.

Буфер канала данных — двуправленный трехстабильный, осуществляет стыковку внутренней шины данных БИС KP580BB79 с системной шиной данных с помощью линий D7... D0.

Схема управления чтением-записью управляет блоками контроллера при приеме информации с системной шины данных и выдаче информации на эту шину. Для управления схемой используются следующие сигналы:

\overline{CS} , \overline{RD} , \overline{WR} — разрешают проведение операций чтения-записи, выдачу информации на системную шину данных, прием информации с системной шины данных соответственно;

INS/\overline{D} — определяет передачу команд и слов состояния ($INS=1$) или передачу данных ($\overline{D}=0$).

Таблица 2

Входные сигналы				Вид операции
INS/\overline{D}	\overline{RD}	\overline{WR}	\overline{CS}	
1	1	0	0	Запись команды
0	1	0	0	Запись данных
1	0	1	0	Чтение слова состояния
0	0	1	0	Чтение данных
X	X	X	1	Микросхема отключена от системной шины

Примечание. X — безразличное состояние.

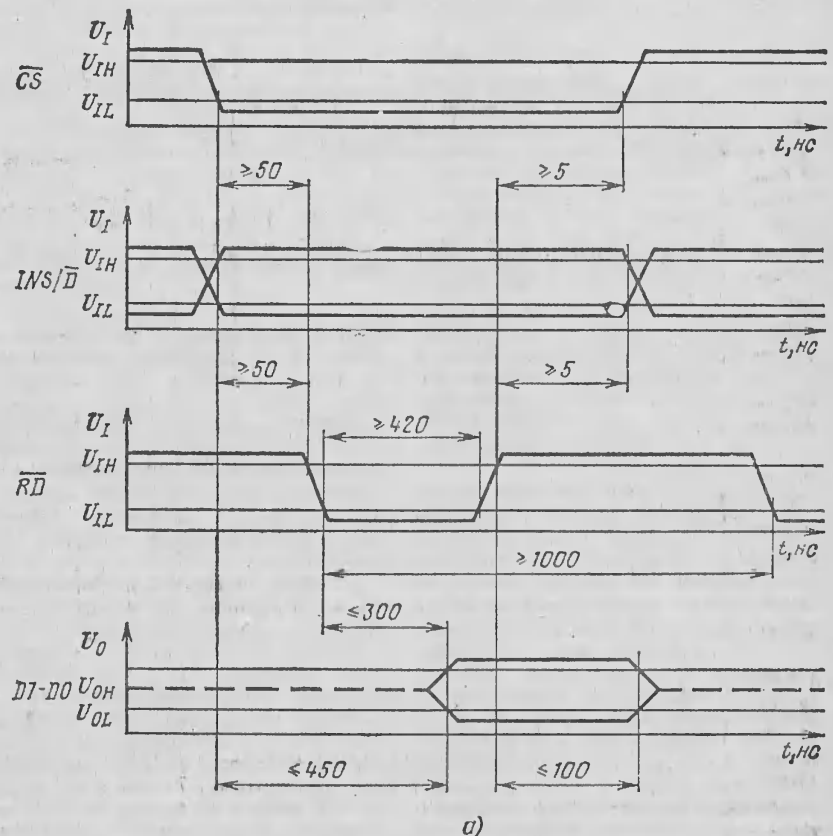


Рис. 3 (начало)

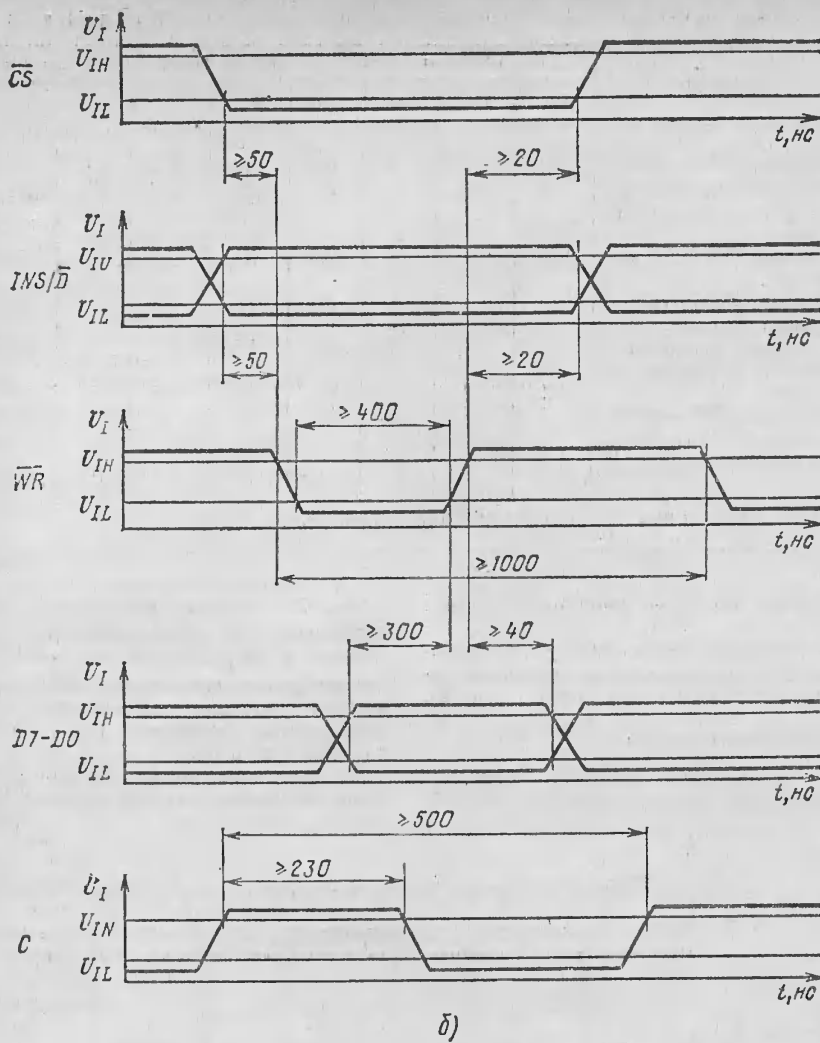


Рис. 3. Временные диаграммы работы микросхемы при чтении данных и слова состояния (а) и при записи данных и команд (б) (окончание)

внутреннюю частоту синхронизации 100 кГц путем деления частоты внешней синхронизации на необходимое число (2...31). Коэффициент деления после аппаратного сброса равен 31, а при программном сбросе не изменяется.

Команда «чтение ОМ ОЗУ датчиков» готовит контроллер для чтения ОМ ОЗУ датчиков предварительной записью этой команды. В режимах сканирования клавиатуры и ввода по стробу флаг автоинкрементирования (A4) и адрес ОМ ОЗУ датчиков (A2, A1, A0) не используются. В режиме сканирования матрицы датчиков адрес A2, A1, A0 выбирает одну из восьми строк ОЗУ. Если A4=1, то каждое последующее чтение будет происходить из следующей строки ОЗУ.

Команда «чтение ОЗУ отображения» готовит контроллер для чтения ОЗУ отображения. Адресные биты A3...A0 выбирают одну из 16 строк

ОЗУ отображения. Если A4=1, то адрес будет инкрементироваться после каждого чтения ОЗУ отображения.

Команда «запись в ОЗУ отображения» готовит контроллер для записи информации в ОЗУ отображения. Адресация ОЗУ отображения и автоинкремент адреса происходят так же, как и после команды «чтение ОЗУ состояния».

Команда «гашение — запрет записи отображения» используется для гашения отображения на выходах DSPA3...DSPA0, DSPB3...DSPB0 в нули, единицы и 16-ричное число 20, а также для запрещения записи в один из полубайтов ОЗУ отображения или оба одновременно: G3 запрещает запись в ОЗУ отображения по входам D7...D4 для канала А; G2 запрещает запись в ОЗУ отображения по входам D3...D0 для канала В; G1 обнуляет выходы DSPA3...DSPA0 для канала А; G0

Таблица 4

Код			Режим клавиатуры
K2	K1	K0	
0	0	0	Кодированное сканирование с обнаружением одноклавишных сцеплений (устанавливается после сброса)
0	0	1	Декодированное сканирование с обнаружением одноклавишных сцеплений
0	1	0	Кодированное сканирование с обнаружением N-клавишных сцеплений
0	1	1	Дешифрованное сканирование с обнаружением N-клавишных сцеплений
1	0	0	Кодированное сканирование матрицы датчиков
1	0	1	Дешифрованное сканирование матрицы датчиков
1	1	0	Ввод по стробу, кодированное сканирование дисплея
1	1	1	Ввод по стробу, дешифрованное сканирование дисплея

Таблица 5

Код		Режим дисплея
K4	K3	
0	0	8—8 битовых символов на дисплее, ввод слева
0	1	16—8 битовых символов на дисплее, ввод слева (устанавливается после сброса SR)
1	0	8—8 битовых символов на дисплее, ввод справа
1	1	16—8 битовых символов на дисплее, ввод справа

обнуляет выходы в DSPB3...DSPB0 для канала В.

Гашение отображения или запрет записи информации выполняется при записи единицы в соответствующий разряд G3...G0 команды «гашение — запрет записи отображения».

Команда «сброс» осуществляет программно общий сброс контроллера, а также сброс слова состояния и запрета прерывания, сброс ОЗУ отображения и выходов DSPA3...DSPA0, DSPB3...DSPB0 в код, который определяется разрядами C3 и C2. Разряд C0 выполняет общий сброс

контроллера без изменения коэффициента деления входной частоты и режима работы.

Разряд С1 устанавливает регистр слова состояния в положение «пусто ОМ ОЗУ датчиков», а также сбрасывает выход INT в клавиатурном и ввод по стробу режимах. Разряды С3 и С2 определяют значение кода, в который будут устанавливаться ОЗУ отображения или выходы DSPA3 ... DSPA0, DSPB3 ... DSPB0, т. е. в нуль, единицу и 16-ричное число 20. Разряд С4=1 сбрасывает ОЗУ отображения, а разряд С4=0 сбрасывает выходы DSPA3 ... DSPA0 и DSPB3 ... DSPB0 в код:

С3	С2	Код
0	0	Все нули
1	0	20 ₁₆ =0010000 ₂
1	1	Все единицы

В течение времени сброса ОЗУ отображения (160 мкс) запись в него запрещается. Одновременно устанавливается старший бит слова состояния, который автоматически сбрасывается после сброса ОЗУ отображения. В режиме сканирования матрицы датчиков при С1=1 регистр адреса ОЗУ датчиков сбрасывается на нулевую строку.

Команда «сброс INT — установка обнаружения ошибок» в режиме кодированного (декодированного) сканирования с обнаружением N-клавишных сцеплений при E=1 программирует контроллер для работы в режиме обнаружения ошибок. Эта команда в кодированном (дешифрированном) режиме сканирования матрицы датчиков переводит вывод INT в состояние «Лог. 0» и разрешает дальнейшую запись в ОЗУ датчиков.

Работа БИС КР580ВВ79

Информация в БИС КР580ВВ79 вводится через линии возврата RET7 ... RET0.

В кодированном (дешифрированном) режиме сканирования с обнаружением одноклавишных сцеплений по нажатию клавиши срабатывает логика подавления дребезга. Другие нажатия клавиш обнаруживаются в течение следующих двух сканирований. Если это единственная нажатая клавиша и ОМ ОЗУ датчиков было пусто, то ее позиция записывается в ОМ ОЗУ датчиков вместе с содержанием линий CO/STB и SH, а сигнал INT устанавливается в состояние «Лог. 1», сообщая центральному микропроцессору (ЦП) о вводе информации в ОМ ОЗУ датчиков из клавиатуры. Если ОМ ОЗУ датчиков было заполнено полностью, то информация с клавиши не будет введена и установится флаг ошибки D5 в слове состояния. В случае одновременного нажатия нескольких клавиш код ни одной из них не будет записан в ОМ ОЗУ датчиков. Код клавиши записывается в ОМ ОЗУ датчиков только одиночным нажатием.

В кодированном (дешифрированном) режиме с обнаружением N-клавишных сцеплений каждое нажатие клавиши обрабатывается независимо от всех остальных. При нажатии клавиши схема подавления дребезга ожидает два цикла сканирования и проверяет достоверность нажатия клавиши. При подтверждении нажатия в двух циклах код клавиши записывается в ОМ ОЗУ датчиков. Если встретится одновременное нажатие, то коды клавиш будут распознаны и введены в соответствии с порядком сканирования клавиатуры сигналами S3 ... S0.

В кодированном (декодированном) режиме с обнаружением N-клавишных сцеплений с помощью команды «сброс INT — установка обнаружения ошибок» можно запрограммировать специальный режим обнаружения ошибок. Распознавание и подавление дребезга клавиши происходит, как в обычном N-клавишном режиме. Если в течение одного цикла подавления дребезга две клавиши будут обнаружены нажатыми, то это распознается как одновременные нажатия и устанавливается в регистре слово состояния разряд D6 — флаг ошибки. Указанный флаг запрещает дальнейшую запись в ОМ ОЗУ датчиков и устанавливает на выводе INT состояние «Лог. 1» (если оно до этого еще не было установлено).

В кодированном (дешифрированном) режимах сканирования клавиатуры символы, вводимые в ОМ ОЗУ датчиков, соответствуют позиции датчика в клавиатуре и значению сигналов SH и CO/STB (неинвертированным). CO/STB — старший значащий бит, SH — следующий. Далее три бита указывают строку, в которой находится нажатая клавиша. Последние три бита и указывают, к какой линии (RET7 ... RET0) подсоединена нажатая клавиша.

В кодированном (дешифрированном) режиме сканирования матрицы датчиков логика подавления дребезга заблокирована и состояние ключей датчиков вводится непосредственно в ОЗУ датчиков. Хотя подавление дребезга не обеспечивается, этот режим имеет то преимущество, что ЦП «знает», как долго датчик находится в замкнутом состоянии и когда он был разомкнут. Клавиатурные режимы могут только отобразить действительное замыкание. Для упрощения программное обеспечение разработчик должен функционально сгруппировать датчики в строки в том формате, в котором ЦП будет их читать. Вывод INT устанавливается в состояние «Лог. 1» (если обнаружено изменение состояния датчика) в конце сканирования матрицы датчиков. В состоянии «Лог. 0» вывод INT переходит после первой операции чтения данных, если перед чтением данных была записана команда «чтение ОМ ОЗУ датчиков», а во время записи коман-

ды разряд слова состояния (D4) находился в состоянии «Лог. 0» или при подаче команды «сброс INT — установка обнаружения ошибок» (E=1).

В кодированном (дешифрированном) режиме сканирования матрицы датчиков данные с линий возврата RET7 ... RET0 вводятся прямо в строку ОЗУ датчиков, соответствующую строке матрицы датчиков, которая сканируется. Таким образом, каждая позиция датчика отображается прямо в позиции матрицы ОЗУ датчиков.

Входы CO/STB и SH в этом режиме не используются. Заметим, что к линиям возврата могут быть подключены не только контакты. Любая логика, фиксирующаяся линиями сканирования, выдает данные на входы RET7 ... RET0.

В режиме «ввод по стробу с кодированным (дешифрированным) сканированием дисплея» данные также вводятся в ОМ ОЗУ датчиков через линии возврата RET7 ... RET0. Данные записываются нарастающим фронтом строб-импульса на линии CO/STB. Они могут приходиться от любой кодированной клавиатуры или матрицы датчиков. Режим 16 (8) — 8 битовых символов на дисплее ввода слева — простейший формат отображения, в котором каждой позиции дисплея соответствует определенная строка (полустрока) в ОЗУ отображения. Адресу 0 в этом ОЗУ соответствует крайний слева символ дисплея, а адресу 15 в 16-разрядном дисплее или адресу 7 в 8-разрядном — крайний справа символ. Вывод символов, начиная с нулевой позиции, вызывает заполнение дисплея слева направо. Символ 17- или 9й будет в этом случае вводиться снова в крайнюю слева позицию дисплея и т. д. Вывод в произвольную позицию дисплея при установке режима вывода слева направо без сдвига информации не приводит к нежелательным побочным эффектам, и результат при этом предсказуем. В этом режиме адресация ОЗУ отображения используется с автоинкрементированием и без инкремента. Инкрементирование способствует увеличению адреса, по которому будет производиться следующая запись, на единицу, и символ появляется в следующей позиции. Без инкремента оба ввода произойдут в один и тот же адрес ОЗУ отображения и позицию дисплея.

Режим 16 (8) — 8 битовых символов на дисплее ввода справа используются в большинстве электронных калькуляторов. Первый введенный символ помещается в правую позицию дисплея. Следующий вводимый символ также помещается в правую позицию дисплея, но после того, как сдвинется все отображение на одну позицию влево. Крайний слева символ при этом сдвигается за край дисплея и теряется. В режиме ввода справа со сдвигом нет прямого соответствия между позицией дисплея и адресом

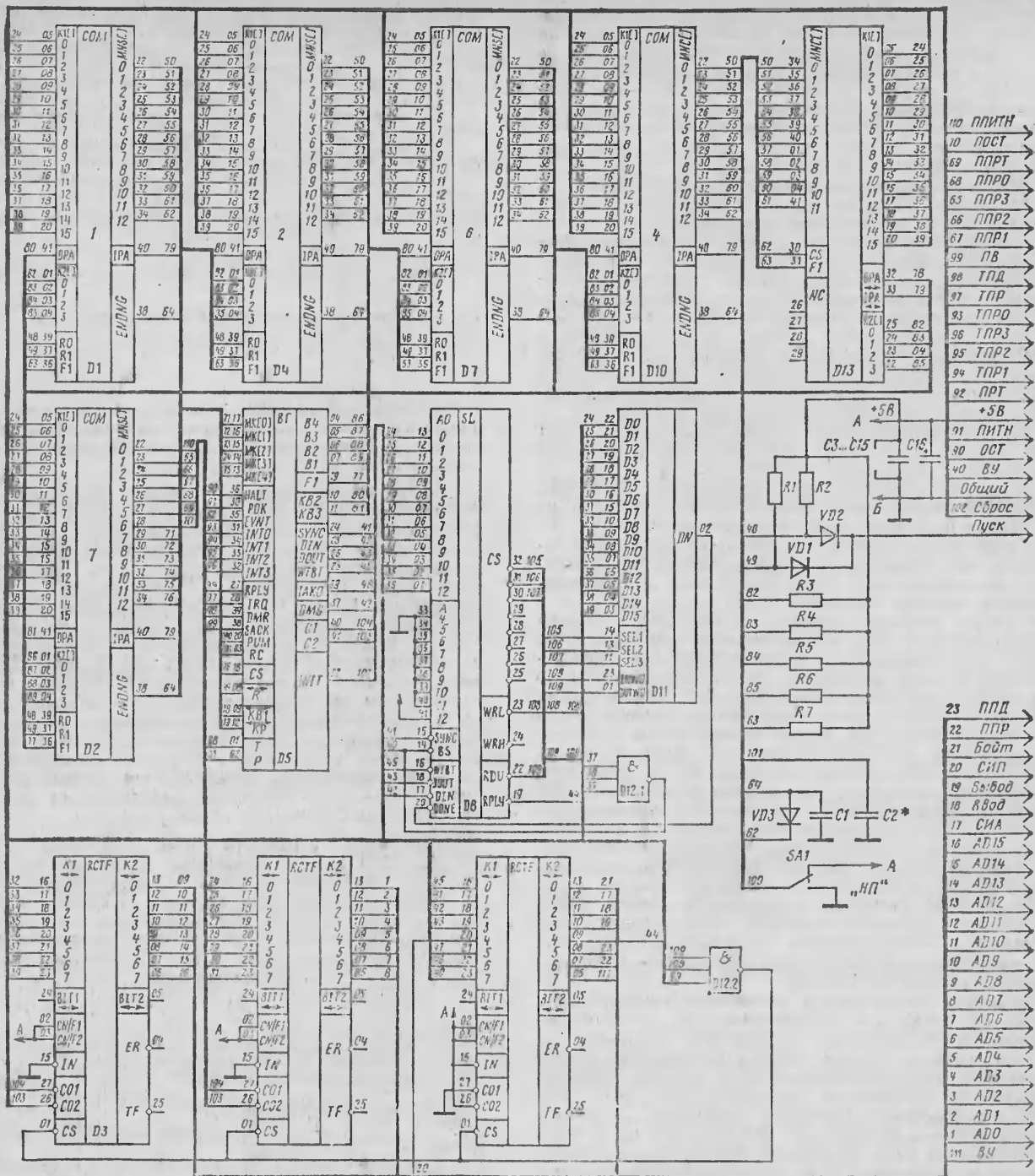


Рис. 2. Принципиальная схема процессора

Основные технические характеристики микроЭЕМ

Производительность, тыс. операций/с	330
Объем ОЗУ, байт	8Кx16
Объем ПЗУ, байт	4Кx16
Потребляемая мощность, Вт	0,6
Напряжение питания, В	5
Габаритные размеры, мм	233x35x220

Процессор. Принципиальная схема процессора приведена на рис. 2. В его состав входят: пять БИС управляющей памяти К/КР588ВУ2 D1, D2, D4, D7, D10 (кодировки 0001, 0002, 0004, 0006, 0007),

составляющие блок микропрограммного управления, предназначенного для формирования микропрограмм в соответствии с кодом выполняемой команды или обслуживаемым прерыванием;

БИС 16-разрядного арифметического устройства D13 К/КР588ВС2 для выполнения логических и арифметических операций над 16- и 8-разрядными словами. Результат операции сопровождается выдачей признаков знака, равенства нулю, переполнения и переноса (N, Z, V, C соответственно). В состав АУ входят 16 регист-

ров общего назначения, из которых 8 (R0...R7) программно доступны. Регистры R0...R5 используются для хранения операндов и промежуточных результатов, а также в качестве адресных или индексных регистров. Регистр R6 служит указателем стека, R7 — счетчиком команд. Остальные 8 регистров используются как служебные при выполнении микропрограмм и недоступны программисту. Текущее слово состояния процессора хранится в регистре состояний БИС АУ.

БИС системного контроллера D5 К/КР588ВГ1, в функции которой входит организация асинхронного обмена данными и командами между процессором и устройствами, подключенными к системной шине, организация обработки внешних или внутренних прерываний, приоритетное кодирование, а также представления системной магистрали внешнему устройству, запрашившему ее для обмена по прямому доступу к памяти;

БИС арифметического умножителя D11 К588ВР2 и БИС селектора адреса D8 К588ВТ1, составляющие блок умножения и предназначенные для аппаратного выполнения операций умножения, на которые, как правило, приходится основные временные затраты;

три БИС магистральных приемопередатчиков (МПП) D3, D6, D9 К/КР588ВА1 для повышения нагрузочной способности процессора.

Следует отметить, что микросхемы К588ИК1, К588ИК2, разработанные ранее специально для станков с ЧПУ, в настоящее время не выпускаются.

Рассматривая принципиальную схему процессора заметим, что линии адреса данных (AD0...AD15) системной магистрали через МПП подключены к двунаправленным выводам информационного канала БИС АУ и ко входам регистра команд всех БИС УП, а выходы микрокоманды БИС УП с кодировками 0001, 0002, 0006, 0004, содержащими микропрограммы АУ, объединяются по схеме «проводное И» и подключаются ко входу регистра микрокоманд БИС АУ. Выдача микрокоманды сопровождается выставлением низкого уровня на вход CS БИС АУ. Прием микрокоманды в регистр микрокоманд БИС АУ завершается выдачей отрицательного фронта сигнала на входе F1. В процессоре реализуется конвейерное выполнение микрокоманд, при котором прием и дешифрация очередной микрокоманды совмещаются с исполнением предыдущей. В кодировке 0001 защиты макропрограммы выполнения основных команд из системы команд микроЭВМ «Электроника 60», в кодировке 0002 — команды ветвления, в кодировке 0006 — команды умножения, деления и много-разрядных сдвигов, в кодировке 0004 — команды прерываний и микропрограммы обслуживания внутренних и внешних прерываний, в кодировке 0007 — микропрограммы управления БИС СК.

Взаимная синхронизация выполнения микропрограмм БИС СК и БИС АУ осуществляется по сигналам квитирования передаваемой информации, которыми АУ и СК связаны между собой. Выходы магистрали состояний БИС АУ соединяются со входами магистрали признаков ветвления кодировок 0001, 0002, 0006, входы магистрали признаков ветвления кодировок 0004, 0007 — с выходами В1...В4 БИС СК, на которых формируются коды прерываний. Входы-выходы ENDNG «конец команды» всех БИС УП объединяются по схеме «проводное И» и предназначены для синхронизации приема команд. БИС УП формирует высокий уровень на выводе ENDNG по приему в регистр микрокоманд БИС АУ последней микрокоманды из микропрограммы выполняемой команды. При этом очередная команда в БИС АУ принимается только при наличии высокого уровня на выходах ENDNG всех БИС УП процессора и низкого уровня на входах ОРА. Выходы квитирования выдачи команды KB2, KB3 БИС СК подключены ко входам ОРА 0001, 0002, 0006 и 0004, 0007 соответственно. Вход-выход квитирования выдачи данных KB1 БИС СК соединяется со входом-выходом ОРА БИС АУ. Выходы квитирования приема ИРА БИС АУ, БИС УП и вывод КР БИС СК объединены по схеме «проводное И».

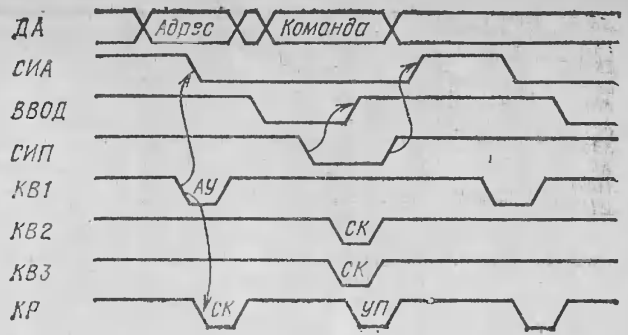
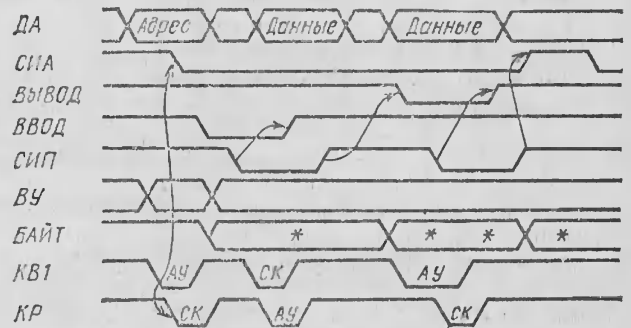


Рис. 3. Временная диаграмма выполнения цикла чтения команды

Временные диаграммы выполнения циклов чтение команды и чтение-модификация-запись данных, поясняющие работу процессора по выводам квитирования, приведены на рис. 3, 4.

Если во время выполнения команды процессор принял прерывание (выставил код на выводах В1...В4), то в цикле приема следующей команды будет выставлен только сигнал KB3 и процессор приступит к обслуживанию принятого прерывания. В данном процессоре реализована многоуровневая приоритетная система прерываний. Вывод HALT БИС СК является входом прерывания, обслуживаемого через адрес-вектор 160002. Вывод EVNT — вход прерывания по таймеру с адресом-вектором обслуживания 100. Вывод POK — вход прерывания по аварии источника питания с адресом-вектором обслуживания 24. При обслуживании этого прерывания БИС СК вырабатывает сигнал каппального сброса INIT. Выводы INT0, INT1, INT2, INT3 — входы прерываний с фиксированными векторами обслуживания 170, 174, 270, 274 соответственно. Вывод IRQ — вход требования прерывания от внешнего устройства с вводимым адресом-вектором.

По выводам процессора ПОСТ, ППИТН, ППРТ, ППРО...ППРЗ выдаются сигналы подтверждения прерываний, формируемые 0007 кодировкой БИС УП. В процессоре реализованы два режима пуска: с адреса 24 и с адреса 173000. Режим выбирается установкой высокого (адрес 24s) или низкого уровня (адрес 173000a) на выводе PUM БИС СК. Выводы P и T



* - уровень сигнала не имеет значения
** - устанавливается при байтовых операциях

Рис. 4. Временная диаграмма выполнения цикла чтение-модификация-запись

В. А. Бобков, Б. Н. Чернуха, Д. Н. Черняковский,
В. С. Свиридович, В. Н. Кушарев

СЕЛЕКТОР АДРЕСА КР588ВТ1

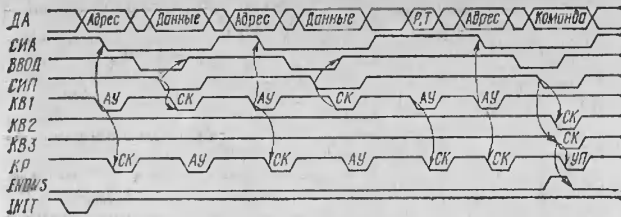


Рис. 5. Временная диаграмма выполнения начальной установки процессора

БИС СК являются входами триггеров Р-бита и Т-бита, которые устанавливаются при загрузке слова состояния процессора.

Для быстрого выполнения команд умножения в состав процессора включены БИС селектора адреса и БИС арифметического умножения. Селектор адреса позволяет организовать загрузку операндов, считывание результата и признаков из арифметического умножителя. Операции умножения выполняются аппаратно. Во время чтения и записи операндов магистральные приемопередатчики должны выключаться при наличии сигналов RDU и WRZ.

Для пуска процессора необходимо подать импульс длительностью не менее 0,5 мкс на вход «пуск». При этом БИС УП 0007 кодировки и БИС УП 0004 кодировки выдают микропрограмму начальной установки, выполнение которой в случае старта с адреса 24₈ состоит из следующих фаз:

- выдача сигнала канального сброса INIT;
- прием счетчика команд (R7) из ячейки с адресом 24₈;
- прием слова состояния процессора из ячейки с адресом 26₈;
- запись значения Р- и Т-битов в БИС СК;
- прием первой команды.

Временная диаграмма начальной установки приведена на рис. 5. В случае старта процессора с адреса 173000₈ начальная установка начинается приемом кода команды из ячейки 173000₈.

ЛИТЕРАТУРА

1. Черняковский Д. Н., Шиллер В. А., Юровский А. А. Процессор с системой команд и интерфейсом микроЭВМ «Электроника 60» на основе БИС серии КР588 // Электронная промышленность, — 1983, — Вып. 9 (126). — С. 11—13.
2. Коннов Е. В., Тихомиров С. Н., Черняковский Д. Н., Шиллер В. А. Модуль ОЗУ с унифицированным интерфейсом на основе БИС серии КР588 // Электронная промышленность, — 1983, — Вып. 9 (126). — С. 14—17.
3. Давыдовская В. В., Малинин А. В., Таратын И. А., Усов Г. И. Быстродействующие асинхронные статические ОЗУ К537РУ14 // Микропроцессорные средства и системы, — 1986, — № 4. — С. 49—50.

Статья поступила 1 июля 1987 г.

Большая интегральная микросхема селектора адреса (БИС СА) К/КР588ВТ1 предназначена для применения в контроллерах внешних устройств (ВУ), подключаемых к системной магистрали по ГОСТ 26765.51-86 (шина МПИ), и используется для управления чтением-записью регистров контроллеров. Микросхему можно также применять в качестве контроллера запоминающего устройства, подключенного к шине МПИ.

БИС СА осуществляет выборку восьми 16-разрядных или шестнадцати 8-разрядных регистров при записи и восьми 16-разрядных регистров при чтении. Микросхема, изготавливаемая на основе низкороговой КМОП-технологии с самосовмещенным поликремниевым затвором, имеет два исполнения:

К588ВТ1 — в металлокерамическом корпусе типа 429.42-5 и КР588ВТ1 — в пластмассовом корпусе типа 2206.42-2. БИС СА состоит из следующих функциональных блоков (рис. 1):

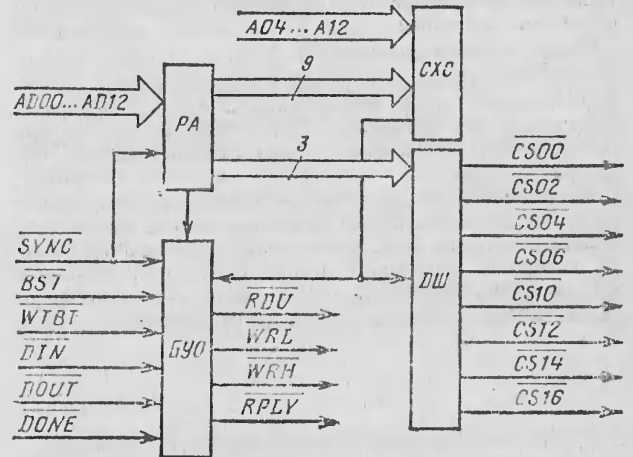


Рис. 1. Структурная схема селектора адреса К/КР588ВТ1

Таблица 1

Зависимость значений сигналов на выводах CS00...CS16 от кода в регистрах PA01...PA03

PA03	PA02	PA01	CS00	CS02	CS04	CS06	CS10	CS12	CS14	CS16
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

блока управления обменом (БУО), осуществляющего управление записью-чтением регистров ВУ;

13-разрядного (РА0...РА13) регистра адреса (РА), хранящего адрес, принимаемый с информационной шины в адресной части цикла обмена;

схемы сравнения (СХС), осуществляющей сравнение 9-разрядного адреса регистров ВУ, задаваемого подключением внешних выводов А04...А12 микросхемы, с соответствующими разрядами адреса, записанного в РА; дешифратора (ДЩ), формирующего сигналы CS00...CS16 в зависимости от информации, находящейся в РА01...РА03, в соответствии с табл. 1.

Условное графическое обозначение микросхемы приведено на рис. 2, назначение выводов показано в табл. 2. Ниже даны основные характеристики ОЗУ.

Основные технические характеристики при T—0...+70°C

Напряжение питания U _{CC} , В	5±5%
Ток потребления в статическом режиме, мА, не более	500
Время задержки сигнала RDU относительно сигнала DIN, нс, не более	180
Время задержки сигналов WRL, WRH относительно сигнала DOUT, нс, не более	225

Временная диаграмма чтения информации из регистров ВУ (цикл «ввод») приведена на рис. 3. Процессор (или другое ведущее устройство) устанавливает на информационной шине адрес ВУ и сигнал выборки ВУ BS7=0, а затем переводит в 0 сигнал SYNC. По сигналу SYNC=0, индицирующему наличие адреса на информационной шине, информация с AD00...AD12 записывается в РА БИС СА. При совпадении данных на выводах А04...А12 с содержимым РА04...РА12 БИС СА активизируется и в соответствии с кодом в РА01...РА03 на одном из выводов CS00...CS16 устанавливается 0. По сигналу DIN=0, поступающему с процессора, активизированная БИС СА формирует сигнал RDU=0, по которому происходит чтение выбранных регистров ВУ на информационную шину, после чего регистры устанавливают сигнал DONE в 0, информируя этим

Таблица 2

Назначение выводов селектора адреса К/КР588ВТ1

Вывод	Назначение
1...13	Входы адреса информационного канала AD12...AD00
14	Вход «выбор ВУ» BS7
15	Вход «синхронизация обмена» SYNC
16	Вход «запись байта» WTBT
17	Вход «чтение» DIN
18	Вход «запись» DOUT
19	Выход «ответ» RPLY
20	Вход «ответ регистра ВУ» DONE
21	Общий 0V
22	Выход «чтение регистров ВУ» RDU
23	Выход «запись младшего байта» WRL
24	Выход «запись старшего байта» WRH
25...32	Выходы «выбор регистров ВУ» CS16...CS00
33...41	Входы «адрес ВУ» А04...А12
42	Напряжение питания U _{CC}

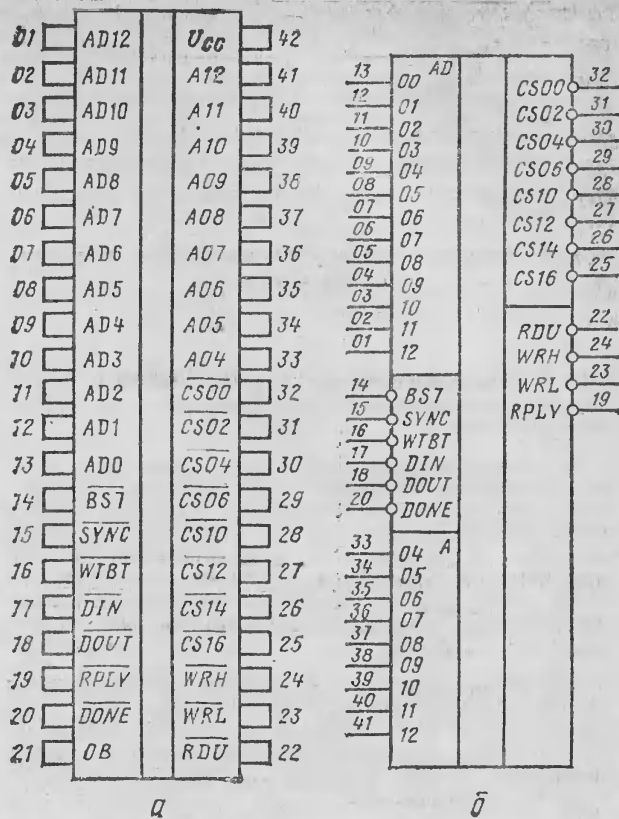


Рис. 2. Условное графическое изображение микросхемы К/КР588ВТ1 по порядку расположения выводов (а) и функциональному назначению выводов (б)

БИС СА о наличии данных на информационной шине. Получая сигнал DONE=0, БИС СА обеспечивает равенство 0 сигнала RPLY, передавая информацию процессору о наличии данных на информационной шине. Получая сигнал RPLY=0, процессор считывает данные с информационной шины и задает равенство DIN=1, по которому БИС СА переводит значение сигналов RDU, DONE в 1. После установки процессором сигнала SYNC в 1 БИС СА переходит в исходное состояние.

Временная диаграмма записи информации во внешний регистр ВУ (цикл «вывод») приведена на рис. 4. Адресная часть цикла «вывод» аналогична циклу

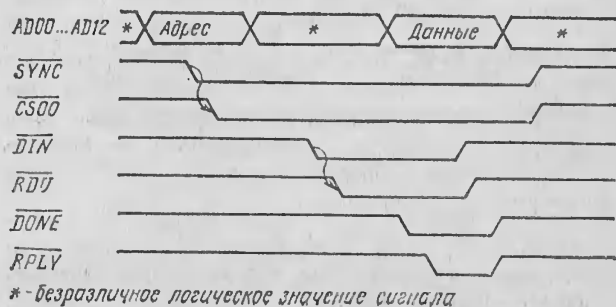


Рис. 3. Режим чтения (цикл «ввод»)

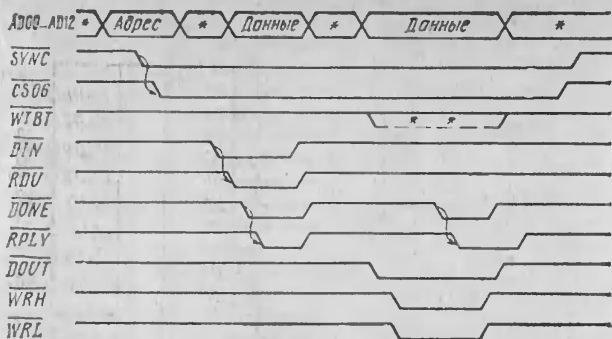
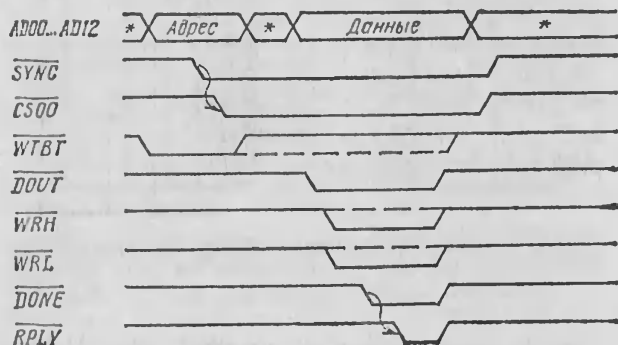


Рис. 4. Режим записи (цикл «вывод»)



Режим 5. Режим чтения-модификация-запись

«ввод», за исключением сигнала \overline{WTBT} , который устанавливается в 0 процессором. По сигналу $\overline{DOUT}=0$ БИС СА обеспечивает равенство нулю сигналов \overline{WRL} и \overline{WRH} , по которым в выбранные регистры ВУ записываются данные (старший и младший байты) с информационной шины, после чего регистры переводят сигнал \overline{DONE} в 0, информируя БИС СА о завершении записи информации. Получая сигнал $\overline{DONE}=0$, БИС СА задает сигнал $\overline{RPLY}=0$, информируя процессор о завершении записи информации регистрами ВУ, и устанавливает в 1 сигналы \overline{WRL} , \overline{WRH} , \overline{DONE} . При записи в регистр ВУ байта информации процессор параллельно с установкой данных на информационной шине формирует сигнал $\overline{WTBT}=0$. Получая сигналы $\overline{WTBT}=0$ и $\overline{DOUT}=0$, БИС СА анализирует содержимое PA00 и при PA00=0 на регистры ВУ поступает только сигнал $\overline{WRL}=0$, а при PA00=1 — только $\overline{WRH}=0$. Таким образом происходит запись старшего или младшего байта информации в выбранный регистр ВУ.

Временная диаграмма чтения информации из регистров ВУ, ее модификации и записи по прежнему адресу приведена на рис. 5 и представляет собой объединение циклов «ввод» и «вывод».

Статья поступила 20 октября 1987 г.

В. Н. Кушарев, В. С. Свиридович, Б. Н. Чернуха,
Д. Н. Черняковский, В. А. Бобков

КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА К ПАМЯТИ К588ВТ2

Контроллер прямого доступа к памяти (КПДП) К588ВТ2 состоит из следующих функциональных узлов (рис. 1):

блока управления вводом-выводом (БУВВ), организующего стандартные циклы обмена с ОЗУ по каналу МПИ;

блока управления (БУ), определяющего работу КПДП;

блока прямого доступа к памяти (БПДП), осуществляющего запрос и захват информационной магистрали в режиме прямого доступа к памяти (ПДП);

буфера ввода-вывода (БВВ), выполняющего электрическое согласование КПДП с информационным каналом МПИ;

16-разрядного суммирующего счетчика слов (СС), контролирующего число слов (байтов), передаваемых в режиме ПДП;

15-разрядного суммирующего счетчика адреса (СА), формирующего последовательность адресов в циклах обмена;

11-разрядного (PC0...PC15) регистра состояния (РС), определяющего режим и условия работы КПДП и имеющего следующие разряды:

PC0 (при PC0=0 ПДП начинается по инициативе ВУ, а при PC0=1 — по инициативе центрального процессора);

PC1 — определяет значение сигнала \overline{SRI} , который по начальной установке КПДП всегда равен 0 (PC0=0);

PC2, PC3 — не используются. При чтении PC PC2=PC3=1;

PC4, PC5 — разряды расширения адреса. Содержимое PC4 определяет значение выхода EXA16, а PC5 — зна-

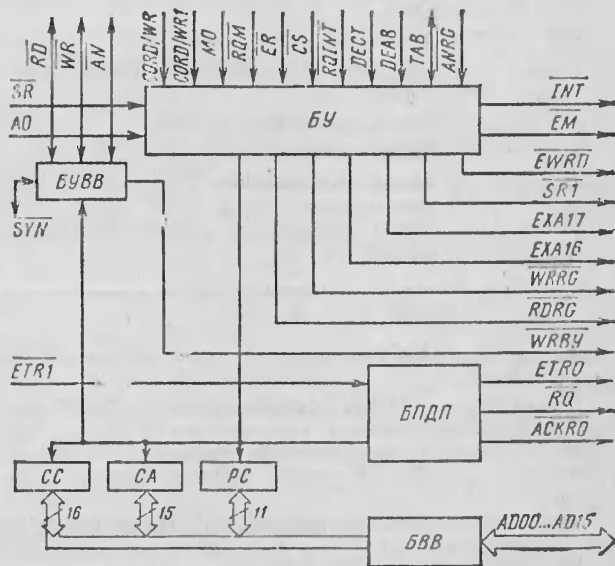


Рис. 1. Структурная схема контроллера К588ВТ2

Таблица 1

Назначение выводов микросхемы К588ВТ2

Вывод	Назначение
1	Выход расширения адреса, EXA17
2	Вход сигнала «режим работы», MO
3, 4	Входы сигналов «управление считыванием-записью» CORD/WR, CORD/WRI
5	Выход сигнала «разрешение записи данных», EWRD
6	Вход «запрет инкрементирования счетчика слов», DECT
7	Вход «запрет инкрементирования счетчика адреса», DEAB
8	Вход адреса данных, AO
9	Выход «запрос магистрали», RQ
10	Вход «разрешение на захват магистрали», ETRI
11	Выход «разрешение на захват магистрали», ETR0
12	Выход «подтверждение запроса», ACKRQ
13	Вход-выход сигнала «ответ устройства», AN
14	Вход «начальная установка», SR
15	Выход «считывание регистра вывода», RDRG
16	Выход «запись регистра ввода», WRRG
17	Вход сигнала «ответ регистра ввода-вывода», ANRG
18	Выход сигнала «признак записи байта», WRBY
19	Вход-выход сигнала «запись», WR
20	Вход сигнала «выборка устройства», CS
21	Вход-выход сигнала «чтение», RD
22	Вход-выход сигнала «синхронизация обмена», SYN
23	Вход-выход «программирование длительности адреса», TAB
24	Напряжение питания, U _{cc}
25...40	Входы-выходы адреса-данных, AD15...AD00
41	Вход сигнала «ошибка», ER
42	Вход сигнала «запрос прерывания», RQINT
43	Выход «установка 1», SRI
44	Выход «разрешение ПДП», EM
45	Выход «прерывание», INT
46	Вход «запрос ПДП», RQM
47	Выход «расширение адреса», EXA16
48	Общий, 0В

чение выхода EXA17 в адресной части циклов обмена информацией;

PC6 — разряд «маски» прерывания. При PC6=0 блокируется выдача сигнала «прерывание» КППД, а при PC6=1 — сигнал «прерывание» выдается КППД по приходу с ВУ сигнала «запрос прямого доступа к памяти»;

PC7 — индицирует состояние КППД. По началу ПДП PC7 устанавливается в 0, а после передачи последнего слова байта PC7=1. Это значение сохраняется и при начальной установке КППД;

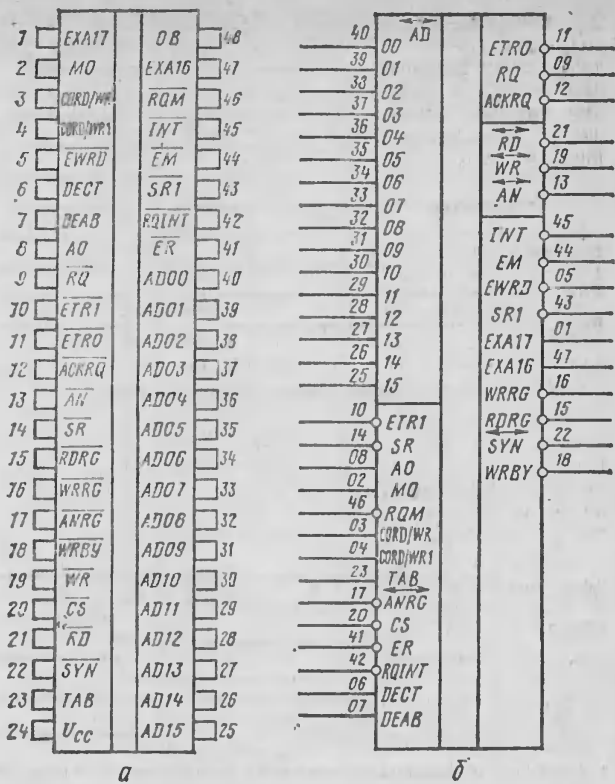


Рис. 2. Условное графическое обозначение микросхемы К588ВТ2 по порядку расположения выводов (а) и по функциональному назначению выводов (б)

PC8 — разрешает работу КППД в режиме ПДП. При PC8=0 переход КППД в режим ПДП запрещен;

PC9, PC10, PC11 — не используются;

PC12 — свободный разряд. Содержимое PC12 определяется данными пользователя;

PC13 — повторяет значение информации на входе «запрос прерывания» КППД;

PC14 — информирует о наличии сигнала на входе «ошибка», с приходом которого PC14 устанавливается в 1. Сброс PC14 осуществляется записью 0 или по начальной установке КППД;

PC15 — сигнализирует о сбое во время ПДП. PC15 = PC14 ∨ PC13.

Условное графическое обозначение микросхем приведено на рис. 2. Назначение выводов показано в табл. 1. Структура устройства, работающего в режиме ПДП (рис. 3), включает селектор адреса К588ВТ1, осуществляющий согласование КППД и ЦП при загрузке или чтении внутренних регистров КППД, магистральные буферные регистры (МБР) К588ВТ1, предназначенные для согласования независимых портов ввода и вывода

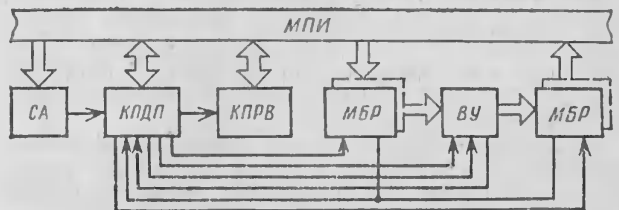


Рис. 3. Пример включения контроллера К588ВТ2

ВУ, контроллер прерываний (КПРВ) K588ВН1, организующий запрос на прерывание, формирование и выдачу адреса-вектора прерывания ВУ и КПДП. Начальная установка осуществляется при включении напряжения питания подачей на вход импульса SR длительностью не менее 500 нс. Ниже приведены основные технические характеристики контроллера.

Технические характеристики контроллера K588ВТ2
 $T = -60 \dots +85^\circ\text{C}$:

Напряжение питания U_{cc} , В	$5 \pm 10\%$
Ток утечки по входам, мкА, не более	20
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
Время задержки сигнала RQM относительно сигнала RO , нс, не более	500

Различают два режима работы контроллера: *однотактовый* — для передачи байта формируется запрос на предоставление ПДП и после его завершения управление магистралью передается ЦП; *монопольный* — данные в режиме ПДП передаются массивом.

Режим работы КПДП определяется значением сигнала на входе MO . При наличии на нем единицы КПДП работает в монопольном, а при наличии нуля — в однотактовом режиме.

КПДП реализует следующие стандартные циклы обмена данными в режиме ПДП:

- «ввод» — чтение информации из ОЗУ;
- «вывод» — запись информации (слова, байта) в ОЗУ;
- «ввод-пауза-вывод» — чтение информации из ОЗУ с последующей записью по данному адресу.

Циклы обмена определяются значениями сигналов, находящихся на входах $CORD/WR$ $CORD/WR1$ КПДП (табл. 2).

Таблица 2

Циклы обмена данными в режиме прямого доступа

Сигнал на входе КПДП		Цикл обмена КПДП в режиме ПДП
$CORD/WR$	$CORD/WR1$	
0	0	Ввод
0	1	Вывод
1	1	Вывод байта
1	0	Ввод-пауза-вывод

Для организации чтения байта из ОЗУ в КПДП предусмотрен вход AO , значение сигнала которого определяет нулевой разряд адреса в циклах обмена $AO = \overline{AD00}$.

Перед началом ПДП ЦП загружает внутренние регистры КПДП, адреса которых определяются разрядами $AD01$ и $AD02$ в адресной части цикла обмена (табл. 3).

Временная диаграмма записи информации во внутренние регистры КПДП приведена на рис. 4. Централь-

Таблица 3

Адреса внутренних регистров контроллера

Сигнал на входе КПДП		Регистр КПДП
$AD01$	$AD02$	
0	0	PC
1	0	CC
0	1	CA



Рис. 4. Режим записи

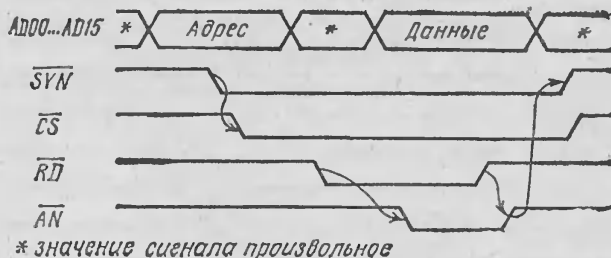


Рис. 5. Режим чтения

ный процессор устанавливает адрес на информационной шине, сопровождая его сигналом $\overline{SYN} = 0$, по которому КПДП записывает и дешифрирует сигналы с $AD01$ и $AD02$, а CA дешифрирует разряды $AD15 \dots AD13$ и задает равенство $CS = 0$, по которому КПДП выбирает один из внутренних регистров. Затем ЦП снимает адрес и устанавливает данные на информационной шине $AD00 \dots AD15$ и сигнал $WR = 0$, по которому КПДП записывает данные во внутренний регистр и обеспечивает

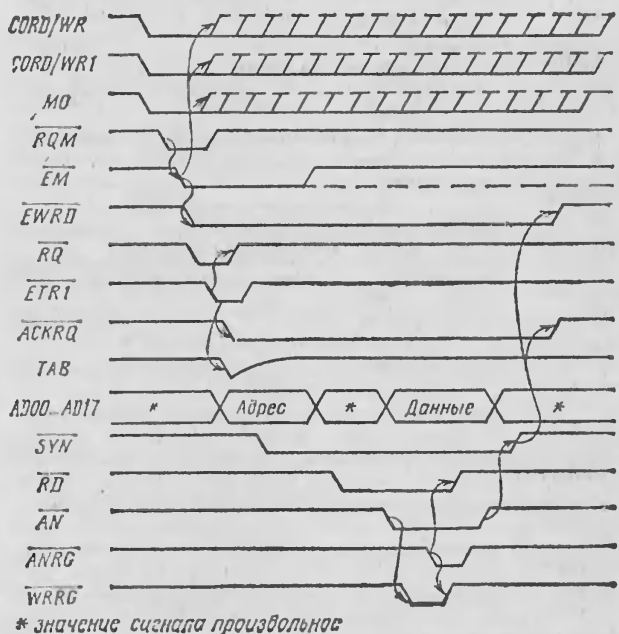
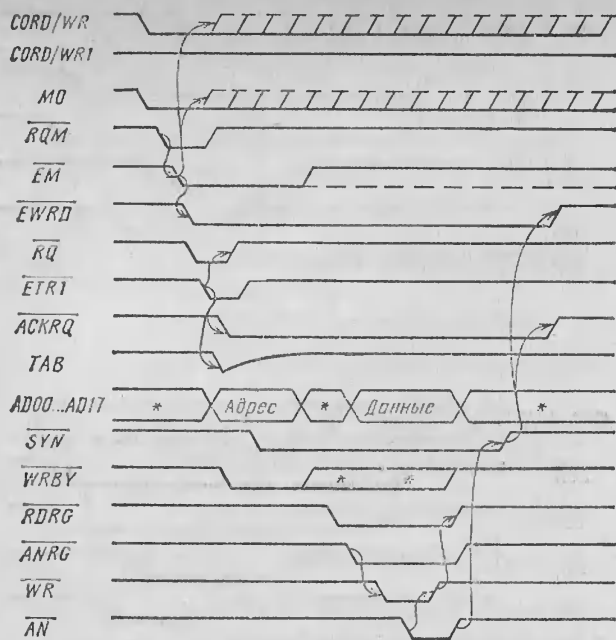
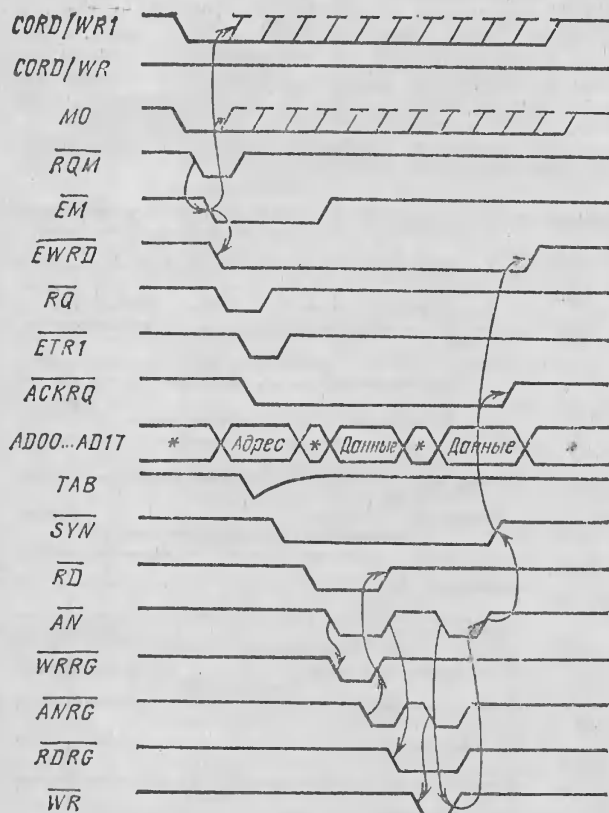


Рис. 6. Временная диаграмма цикла «ввод» в однотактовом режиме



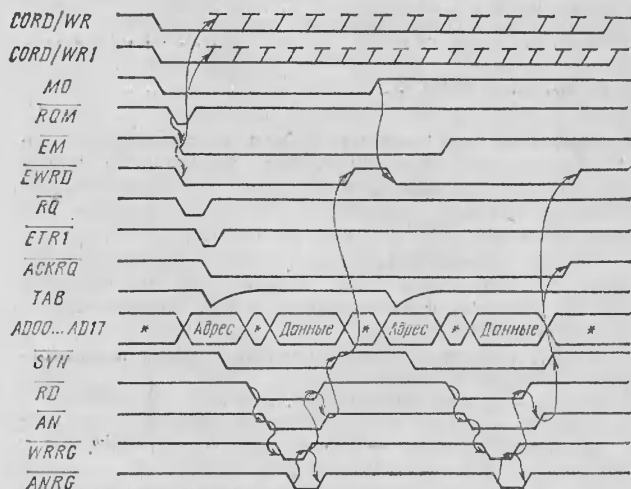
* значение сигнала произвольное
 ** значение сигнала при байтовом обмене

Рис. 7. Временная диаграмма цикла «вывод» в одноктактовом режиме



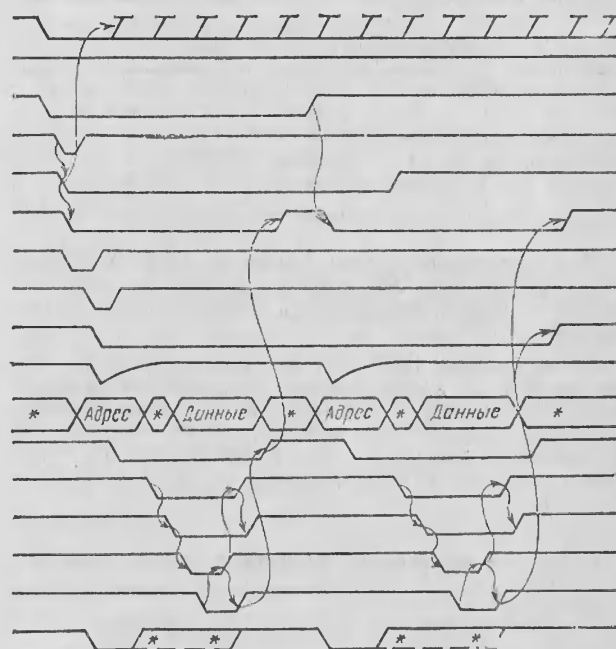
* значение сигнала произвольное

Рис. 8. Временная диаграмма цикла «ввод-пауза-вывод» в одноктактовом режиме



* значение сигнала произвольное

Рис. 9. Временная диаграмма цикла «ввод» в монопульсном режиме



* значение сигнала произвольное
 ** значение сигнала при байтовом обмене

Рис. 10. Временная диаграмма цикла «вывод» в монопульсном режиме

равенство $\overline{AN}=0$, сообщая этим ЦП об окончании записи.

Временная диаграмма чтения информации из внутренних регистров КППД приведена на рис. 5. Адресная часть режима чтения информации из регистров КППД аналогична режиму записи. После снятия адреса ЦП вырабатывает сигнал $\overline{RD}=0$, по которому КППД устанавливает на вывотах $AD00...AD15$ содержимое внутреннего регистра и сигнал $\overline{AN}=0$. Получая $\overline{AN}=0$, ЦП считывает данные и переводит сигнал \overline{RD} в 1. Выдача данных КППД прекращается.

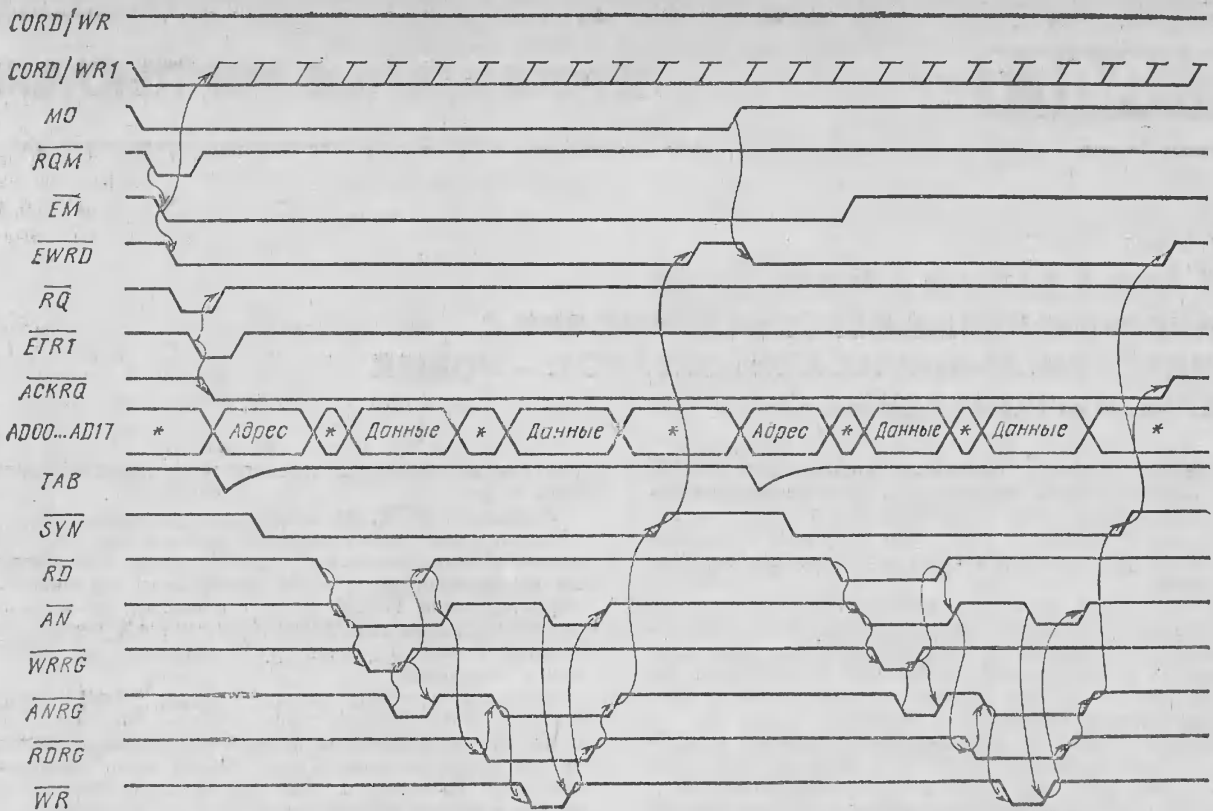


Рис. 11. Временная диаграмма цикла «ввод-пауза-вывод» в монопольном режиме

Временная диаграмма цикла «ввод» приведена на рис. 6. Перед началом работы на выводах CORD/WR, CORD/WR1, MO формируются значения сигналов, определяющие необходимый режим и цикл работы. Длительность нахождения адреса на информационной шине AD определяется параметрами RC-цепи, подключенной к выводу TAB. После снятия адреса ячейки по сигналу $\overline{RD}=0$ на информационной шине AD00...AD15 устанавливается содержимое ячейки ОЗУ и сигнал $\overline{AN}=0$, подтверждающий достоверность данных. Получая сигнал $\overline{AN}=0$, КППД выдаст сигнал $\overline{WRRG}=0$, по которому данные с информационной шины AD записываются в МБП ввода. Затем МБП вырабатывает сигнал $\overline{ANRG}=0$. Контроллер, получая этот сигнал, задает сигналы \overline{WRRG} и \overline{RD} , равные 1, а МБП переводит сигнал \overline{ANRG} в 1. ОЗУ, получая сигнал $\overline{RD}=1$, снимает данные с информационной шины AD и обеспечивает равенство $\overline{AN}=1$, в соответствии с которым КППД устанавливает сигналы \overline{SYN} , \overline{ACKRQ} и \overline{EWRD} в 1, освобождая системную магистраль и сообщая ВУ о наличии данных в МБП и окончании цикла ПДП.

Сигнал \overline{EM} формируется КППД при обнулении CS по снятию адреса с информационной шины AD или сохраняет свое значение до цикла, в котором произойдет обнуление CS.

Временная диаграмма цикла «ввод» приведена на рис. 7. Начало цикла аналогично временной диаграмме цикла «ввод», за исключением того, что параллельно с адресом ячейки ОЗУ КППД устанавливает сигнал \overline{WRBU} в 0, который при записи слова в ОЗУ пере-

дится в 1 при снятии адреса ячейки ОЗУ с информационной шины, а при записи байта в ОЗУ находится в состоянии 0 в течение всего цикла записи.

После снятия с информационной шины AD адреса ячейки ОЗУ контроллер вырабатывает сигнал $\overline{RDRG}=0$, по которому МБП вывода устанавливает данные на информационной шине AD00...AD15 и сигнал $\overline{ANRG}=0$, информирующий КППД о достоверности данных на информационной шине AD. Получая сигнал $\overline{ANRG}=0$, КППД обеспечивает равенство $\overline{WR}=0$, по которому данные с информационной шины AD записываются в ОЗУ. Затем ОЗУ формирует сигнал $\overline{AN}=0$, информируя этим КППД об окончании записи. Контроллер, получив этот сигнал, устанавливает сигналы \overline{WR} и \overline{RDRG} в 1, а МБП — снимает данные с информационной шины и переводит сигнал \overline{ANRG} в 1. ОЗУ по сигналу $\overline{WR}=1$ задает сигнал $\overline{AN}=1$, по которому КППД устанавливает сигналы \overline{SYN} , \overline{ACKRQ} , \overline{EWRD} в 1, освобождая системную магистраль и сообщая ВУ об окончании цикла ПДП.

Цикл «ввод-пауза-вывод» представляет собой объединение циклов «ввод» и «вывод». Временная диаграмма цикла «ввод-пауза-вывод» приведена на рис. 8.

Монопольный режим работы отличается от одноклового тем, что контроллер не освобождает системную магистраль после цикла обмена, а по сигналу MO начинает новый цикл, и так до тех пор, пока CS не обнулится. Временные диаграммы работы КППД в монопольном режиме приведены на рис. 9—11.

Статья поступила 20 октября 1987 г.

УДК 681.322.1

В. И. Иванов, А. В. Клепов, Д. В. Костиков

МАЛОГАБАРИТНЫЕ ПЕРСОНАЛЬНЫЕ ЭВМ С ЖИДКОКРИСТАЛЛИЧЕСКИМ ЭКРАНОМ — НОВЫЙ КЛАСС ПЕРСОНАЛЬНЫХ ЭВМ

Совершенствование технологии производства КМОП ИС высокой степени интеграции и экономичных жидкокристаллических (ЖК) экранов большой информационной емкости привело к появлению малогабаритных персональных компьютеров (МПК) с автономным электропитанием.

К настоящему времени в капиталистических странах продаются около сотни моделей таких ЭВМ, существенно различающихся по техническим характеристикам, стоимости и выпускаемых более чем 50 фирмами. Их разделяют [1—3], как правило, на два класса по размерам, массе и потребляемой мощности (табл. 1).

Персональные ЭВМ с ЖК-экранами ручные (HAND HELD) или карманные (POCKET) МПК (РМПК) стоят 70 ... 400 долл. (без учета используемого совместно с ННПС периферийного оборудования и сменных модулей памяти). Отметим, что стоимость МПК — не определяющий фактор при их классификации и постоянно снижается [4].

Портфельные МПК (BRIEFCASE) или для работы на коленях (LAP-SIZE, KNEETOP) МПК (ПМПК) стоят не более 4000 долл.

В свою очередь ПМПК в большинстве обзоров делятся на две группы: упрощенные портфельные (Lowend) или блокнотные (Notepad) ПМПК. Информационная емкость их ЖК-экранов сравнительно небольшая (в одной строчке не более 40, а общее число не более 640 знакомест). Полнофункциональные (Full function) МПК с характеристиками, приближающимися к подклассу портативных настольных персональных ЭВМ (ПЭВМ) (Suitcase), обычно с ЖК-экранами полного размера (25×80 символов) и клавиатурой, имеющей не менее 83 клавиш (параметры кла-

виатуры и индикатора соответствуют характеристикам IBM PC).

Разделение МПК на классы по размерам, массе, характеристикам экрана вызвано, видимо, тем, что физические и психофизические характеристики пользователя по взаимодействию с ЭВМ практически постоянны, а характеристики ПЭВМ быстро меняются по мере совершенствования технологии. Отметим, что в ряде публикаций можно встретить другие варианты классификации и названий.

Наиболее известны модели фирмы Tandy, Hewlett Packard, Data General, Sharp, Epson, Toshiba, Casio и др. Большинство этих фирм — традиционные изготовители калькуляторов (Casio, Epson) либо персональных ЭВМ настольного типа (Tandy, Data General) либо тех и других одновременно (Hewlett Packard).

Согласно оценке фирмы System Concepts 12 млн. работников только в Западной Европе нуждаются по профилю своей работы в портативных ЭВМ с автономным электропитанием. Наибольший успех имеют ПМПК, сочетающие относительно малые размеры и стоимость с большими функциональными возможностями, практически не уступающими характеристикам настольных ПЭВМ и превосходящие РМПК по удобству пользования. Ожидается ежегодное удвоение их производства вплоть до 1990 г., в частности в 1987 г. их количество превысило 1,7 млн.

Ручные МПК (РМПК)

Области применения ручных МПК, несмотря за более чем трехлетний период их существования, только начинают складываться. Первоначально их использо-

Таблица 1

Сравнение характеристик разных классов МП с ЖК-экраном

Класс	Процессор (разрядность)	Память ОЗУ, К байт	Емкость экрана	Операционная система	Размеры, мм	Масса, кг	Длительность работы от встроенных источников тока, ч
HAND HELD PC	4 ... 8	8 ... 16	16×1 (40×2)	Нет	240×140×30	1,2	30 ... 300
LAP SIZE PC	8	16 ... 64	Не более 40×16	Нет	300×220×40	Не более 3	До 20
FULL FUNCTION	16 применяется сопроцессор	64 ... 768	80×25 (80×16)	Есть	480×350×100	Не более 7	До 8

Малогобаритные ЭВМ с ЖК-экраном. Класс: РМПК

Фирма	Модель	Электропитание		Габариты (длина × ширина × высота), мм	ОЗУ стандартное, К байт	ОЗУ максимальное, К байт
		длительность работы от одного комп- лекта батарей	Наличие сетевого источника питания			
CANON	5000	8	—	257×100×50	16	96
CANON	X-07	*	*	200×130×30	8	42
CASIO	FX-700	300	—	153×69×10	2	—
CASIO	FX-702P	240	—	165×82×17	2	—
CASIO	PB-110	360	—	165×71×9,8	*	*
CASIO	PB-700	100	—	200×88×23	4	16
HEWLETT- PACKARD	HP-71B	*	+	188×100×12,5	17,5	33,5
HEWLETT- PACKARD	HP-71B	*	+	188×100×12,5	17,5	33,5
RADIO- SHARK	PC-2	50	—	193×88×25	1,9	9,9
SHARP	PC-1250	300	*	175×70×18	4	—
SHARP	PC-1260	*	+	135×94×25	9,9	24
SHARP	PC-1500A	50	*	193×94×25	9,9	24

Окончание табл. 2

Фирма	Модель	ПЗУ, К байт	Накопитель	Число клавиш	Печать	Интерфейс (тип)	Число зна- ком- мест в строке × число строк	Масса, кг	Цена, дол.	Литература	Примечание
CANON	5000	*	—	*	+	1,4	20×2	0,72	410	[35, 37 41, 76]	
CANON	X-07	24	*	65	+	1,2	20×4	0,3	325	[39, 57]	
CASIO	FX-700	*	2	55	+	5	12×1	0,13	100	[124, 134]	
CASIO	FX-702P	*	1	65	*	*	20×1	0,176	*	[13]	
CASIO	PB-110	*	—	54	+	5	12×1	0,119	*	[30]	
CASIO	PB-700	4	2	58	+	2	20×4	0,315	161		
HEWLETT- PACKARD	HP-71B	64	—	55	+	—	22×1	0,3	550	[100]	
HEWLETT- PACKARD	HP-71B	64	—	55	+	—	32×1	1,1	*	[109]	96К
RADIO- SHARK	PC-2	16	2	54	+	1,5	26×1	0,4	239	[134]	16К
SHARP	PC-1250	2,2	2	*	+	5	24×1	0,2	110	[46]	
SHARP	PC-1260	16	2	56	+	5	26×1	0,5	220	[45, 58]	
SHARP	PC-1500A	16	2	56	+	1, 2, 3, 4, 5	26×1	0,5	220	[45, 58, 79, 81]	

Примечания: I. Практически все ЭВМ данного класса оснащаются микропроцессором собственной разработки. Программное обеспечение состоит из зашитой в ПЗУ системы программирования на языке Бейсик. II. Типы дополнительных запоминающих устройств: 1 — встроенный накопитель на магнитных картах, 2 — внешний накопитель на микрокассетах. III. Типы интерфейсов: 1 — последовательный (RS-232, RS-232C), 2 — параллельный (Centronics), 3 — интерфейс подклю-

чения бытового телевизора, 4 — встроенный модем, 5 — интерфейс микрокассетного накопителя, 6 — интерфейс HP1B фирмы Hewlett Packard, 7 — акустический модем. IV. Все модели имеют встроенные гальванические источники электропитания, в основном на литиевых элементах СР-2032 или никель-кадмиевых аккумуляторах. V. Цена изделий приводится по литературным сведениям. VI. Принятые обозначения: * — нет сведений, + — имеется в наличии, — отсутствуют.

вали в качестве учебных тренажеров для обучения программированию. По мере совершенствования технических характеристик РМПК (табл. 2) внедряются во многие области человеческой деятельности, где ранее либо вообще не было подходящих средств автоматиза-

ции, либо применялись настольные персональные ЭВМ (ПЭВМ).

Ручные МПК применяются в обучении программированию (первоначальном), научно-технических расчетах, плановых и экономических расчетах, играх, а также в

качестве связанного терминала для работников почты, страхового агентства, устройства для проведения тестирования электронного оборудования.

Архитектура РМПК (рис. 1) в значительной степени унаследована от персональных ЭВМ. В то же время на схемотехнику и конструкцию РМПК в большей мере повлияли технические решения, характерные для калькуляторов. Вместо чисто шинной структуры персональных ЭВМ в РМПК, как правило, используются специализированные интерфейсы подключения к микропроцессору (МП) внешних устройств (клавиатуры, индикатора, сигнализатора). Отметим, что в различных РМПК, выпускаемых одной фирмой, применяется постоянная номенклатура ИС, например, фирмы Sharp и Casio в большинстве своих моделей используют одни и те же БИС, а функциональные возможности (при переходе к новым моделям) нарастают увеличением объемов ОЗУ и ПЗУ, подключением дополнительных БИС интерфейсов и совершенствованием ПО, зашитого в ПЗУ. Часто пользователь РМПК может подключить дополнительные сменные модули памяти. У ряда моделей в ПЗУ зашиты специальные программы обработки вводимой информации (для помощи торговому, страховому работнику и др.).

Все микросхемы, используемые в РМПК, изготавливаются по КМОП-технологии. Это позволяет обеспечить 30...300 ч работы ЭВМ от встроенных источников тока. Для отображения обрабатываемой информации применяется матричный ЖК-экран (информационная емкость может изменяться от 16 до 80 знакомест, расположенных в одну-две строки). Это создает известные неудобства пользователю при обработке информации. Некоторые РМПК имеют встроенные или внешние интерфейсы сопряжения с бытовыми телевизорами (например, модель 40 фирмы Texas Instrument).

В клавиатурах РМПК в основном сохраняется стандартное (как у персональных ЭВМ обычного размера) расположение знаков. Это облегчает переход пользователей от стандартных ЭВМ (микро-, мини-ЭВМ) к РМПК и наоборот. В некоторых типах РМПК применяются дополнительные клавиши для выполнения функций калькулятора (например, HP-71B).

Практически все РМПК имеют те или иные интерфейсы для сопряжения с дополнительными внешними устройствами. Фирмы-изготовители разрабатывают и выпускают специально для применения с РМПК набор микропериферийных устройств. Эти миниатюрные принтеры, плоттеры, микрокассетные накопители отличаются дешевой и малым потреблением энергии, что позволяет им работать от встроенных источников тока.

Благодаря интерфейсам RS-232 и RS-232C к РМПК подключается стандартное периферийное оборудование. РМПК можно также использовать в качестве терминалов к ЭВМ более высокого уровня. В ряде моделей (например, HP-75C) информацию можно передавать через встроенный акустический модем (200...300 Бод).

Конструкция РМПК в основном определяется размерами индикатора, расположением клавиатуры, а так-

же периферийным оборудованием. Наиболее часто используется конструкция РМПК в виде моноблока, к которому в случае необходимости подключается микропериферийное оборудование. Характерны следующие конструктивные принципы подключения периферийного оборудования: через соединители, разъемы (Canon, Sharp); в особом объединяющем конструктиве Casio (объединение РМПК и внешних устройств); к основной конструкции РМПК с помощью кабелей (HP-75C).

Программное обеспечение РМПК рассчитано на пользователей, ранее не работавших с вычислительной техникой. Поэтому в большинстве типов таких МПК используется хранимый в ПЗУ интерпретатор стандартного языка программирования Бейсик. Для подключения внешних микропериферийных устройств и установления связи по телефонному каналу разработаны специализированные программы, обеспечивающие необходимые протоколы обмена. Для ряда моделей РМПК на языке Бейсик разработаны различные игровые программы, которые хранятся, как правило, на внешних магнитных носителях, с которыми может работать ПЭВМ, или вводятся пользователем с клавиатуры. Имеются также демонстрационные примеры и программы для научных расчетов.

Особенности портативных (портфельных) МПК

Основные области применения портативных МПК (ПМПК) типа «Briefcase» были определены фирмами-изготовителями. Например, Tandy Corporation (перед производством наиболее массового в настоящее время ПМПК «Briefcase» Tandy 100 (144)) предполагала использование своей ЭВМ в качестве системы обработки текстовой информации; связанного терминала (Telecommunications); автоматизированного справочника (Information Management); средства для проведения научно-технических расчетов; базы данных (Date Base Management); средства табличных вычислений (Spread Sheet Analyses); интеллектуального терминала к более мощной ЭВМ; устройства для тестирования электронного оборудования; игротехники; контроллера управления бытовыми приборами.

Отметим, что эти области применения характерны для простых моделей ПМПК (группа Lowend), появившихся на рынках в 1982—1983 гг. Более совершенные (дорогие) полнофункциональные портативные ЭВМ могут применяться практически везде, где используются обычные настольные ПЭВМ.

Возможности РМПК в большинстве случаев недостаточны для решения указанных задач из-за неудобства миниатюрных клавиатур и малой информационности ЖК-экрана.

Многие схемотехнические и конструктивные решения ручных МПК были использованы для создания портфельных ЭВМ, однако, определив области применения ЭВМ данного типа, фирмы-изготовители особое внимание уделили:

- применению стандартных операционных систем;
- увеличению скорости работы МПК за счет применения процессоров с высокой тактовой частотой (до 5 МГц) и большей разрядностью (16) и периферийных сопроцессоров;
- совершенствованию эргономических показателей ЭВМ особенно за счет применения полноформатной клавиатуры;
- увеличению информационной емкости экрана;
- повышению объема памяти ОЗУ и ПЗУ, включению в состав МПК дисководов, электронных дисков (ЭД) большой емкости (табл. 3).

В ЭВМ группы «Briefcase» используются в основном процессоры со стандартной шиной (например, Multibus), как правило, КМОП-аналоги приборов фирмы Intel или Zilog.

В ПМПК группы Lowend (упрощенных) наиболее часто используются 8-разрядные, в более дорогих

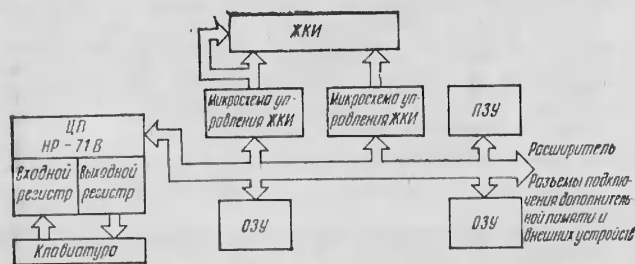


Рис. 1. Упрощенная структурная схема ЭВМ-HP-71B фирмы Hewlett-Packard

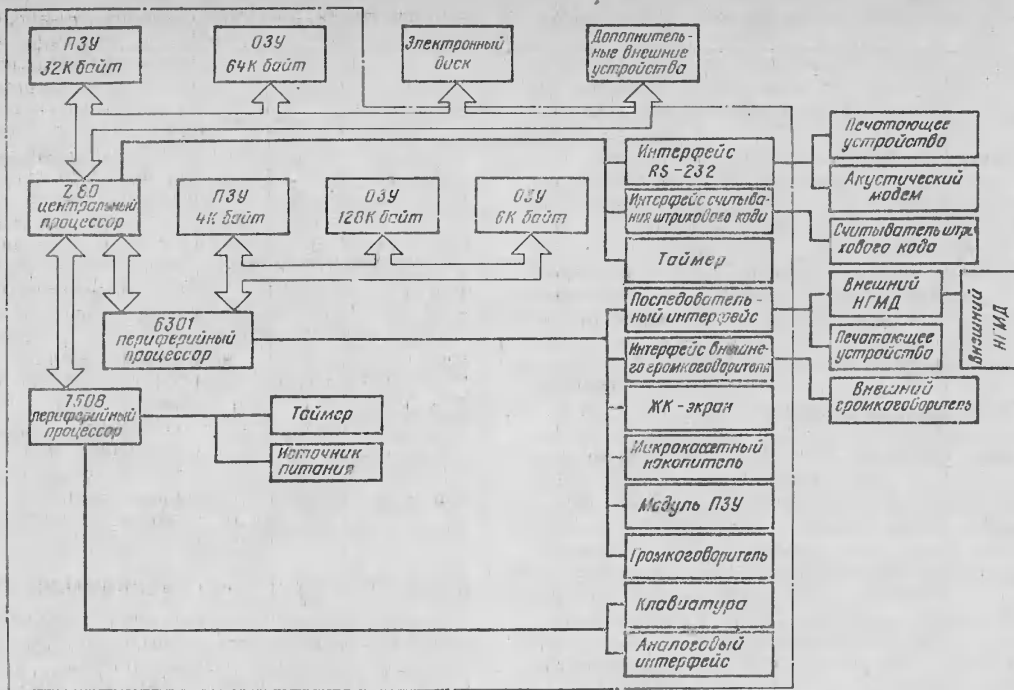


Рис. 2. Структурная схема персональной ЭВМ РХ-8

Таблица 3

Малогабаритные ЭВМ с ЖК-экраном. Класс: РМПК

Фирма-изготовитель	Модель	Габариты (длина × ширина × высота), мм	Масса, кг	Система программирования	Тип процессора	Число знаков × число строк экрана	ЭУ, К байт	
							ОЗУ стандарт	ОЗУ макс. емкостное
AMPERE	ALP	295×265×73	2,4	MS-DOS	HD68000	80×16	64	256
CASIO	FP-200b	310×220×55,5	1,54		80c85	20×8	8	32
APPLE	IIc	300×268×36	3		65S02	80×25	128	*
COMMODORE	COMMODORE	250×263×86	1,2	Собственная	*	80×16	32	*
DATA GENERAL	DG-ONE	250×300×75	3,6	MS-DOS	80C88	80×25	128	512
EPSON	PX-4	295×286×35	1,6	CP/M-86	Z-80 (CMOS)	40×8	64	208
EPSON	PX-8	288×213×33	2	CP/M	Z-80	80×8	64	128
GRID	GRID	380×290×57	5,4	MS-DOS	80c86	80×25	128	512
HEWLETT-PACKARD	HP-110	325×250×75	4,4	MS-DOS	80c86	80×16	96	256
HEWLETT-PACKARD	HP-PLUS	325×250×25	4	MS-DOS	80c86	80×25	128	894
KAYPRO	2000	320×280×70	5	MS-DOS	8088	80×25	256	640
NEK	8401A	300×213×62,5	1,8	CP/M	80c87	80×16	64	*
RADIO-SHARC	200	293×213×51	1,8	Собственная	80c85	40×16	24	72
SORD	IS-11A	300×113×28	1,8	Собственная	Z-80	40×8	32	64
COMPUTER	IS-11C	310×200×75	2,4	MS-DOS	Z-80 (SMOS)	80×25	72	144
SORD	IS-11C	310×200×75	2,4	MS-DOS	Z-80 (SMOS)	80×25	72	144
TOSIBA	T-1100	311×305×66	4,1	MS-DOS	80C88	80×25	256	512
IBM	CONVERTIBLE	*	6	PS-DOS	80c88	80×25	512	1400

Фирма-изготовитель	Модель	ПЗУ, К байт	Дополни- тельные устройства	Число клавиш	Интерфейс (типы)	Электро- питание		Цена, дол.	Литература	Примеча- ние
						время работы, ч	наличие сетевото питания			
AMPERE	APL	128	1,6	67	1, 2, 4	+	+	2085	[20, 50]	
CASIO	FP-200b	8	2,6	69	5, 7	+	+		[30, 36, 111]	
APPLE	11C	16	2,6	63	1,3	*	*	3760	[12, 36, 43, 62]	
COMMODORE	COMMODO- RE	96	2,6	72	1, 2, 4	+	+	600	[5, 91, 97]	
DATA GENERAL	DG-ONE	32	2, 4, 6	79	1, 2, 3, 4, 6	8	+	2895	[34, 36, 37, 47]	
EPSON	PX-4	64	5, 6	76	1, 2, 4, 6	+	+	1400	[54]	256К ЭД
EPSON	PX-8	32	2, 5, 6	72	1	20	+	995	[10, 36, 51, 89]	
GRID	GRID	*	2, 3, 6	83	1, 2, 3, 4, 6	5	+	2975	[16, 37]	
HEWLETT- PACKARD	HP-110	384	2,6	83	1, 4, 6	20	*	2895	[21, 36, 45, 71]	
HEWLETT- PACKARD	HP-PLUS	3000	2,6	83	1, 3, 4, 6	20	—	2993	[26, 66, 69]	
KAYPRO	2000	*	4,6	83	1	4	+	1985	[38, 96]	
NEK	8401A	96	1, 2, 6	73	1, 2, 3, 4	10	+	999	[22, 37, 68, 71]	
RADIO-SHARC	200	72	5,6	72	1, 2, 4	16	+	*	[8, 27, 67, 71]	
SORD	IS-11A	76	1, 2, 6	76	1, 2, 3	8	+	735	[44, 65, 98, 113]	60К ЭД
COMPUTER	IS-11C	80	1, 2, 6	76	1, 2, 4, 6	2,5	+	1495	[36, 112]	64К ЭД
SORD	IS-11C	80	1, 2, 6	76	1, 2, 4, 6	2,5	+	1495	[36, 112]	
COMPUTER	IS-11C	80	1, 2, 6	76	1, 2, 4, 6	2,5	+	1495	[36, 112]	
TOSIBA	T-1100	321	3,6	83	1, 2, 6	8	+	2438	[28, 30, 37, 58]	
IBM	CONVER- TIBLE	*	1, 2, 4, 6	78	1, 2, 3, 4, 6	10	+	3000	[142]	

Примечания: I. Типы дополнительных запоминающих устройств: 1 — встроенный микрокассетный накопитель, 2 — внешний НГМД, 3 — встроенный НГМД, 4 — двойной встроенный НГМД, 5 — внешний кассетный накопитель, 6 — устройство печати. II. Типы интерфейсов: 1 — RS-232 (RS-232C), 2 — Centronics, 3 — интерфейс для подключения бытового телевизора, 4 — встроенный модем (акустический); 5 — интерфейс микрокассет-

ного накопителя, 6 — интерфейс HP1В фирмы Hewlett-Packard, 7 — интерфейс для подключения внешнего модема. III. Габариты и цена приводятся с учетом подключенных внешних устройств, размещаемых в одном портфеле с ЭВМ. IV. Принятые обозначения: * — нет сведений, + — имеется в наличии, — отсутствует. V. ЭД — электронный диск (внешнее энергонезависимое ОЗУ).

«Full Function» применяются 16-разрядные МП, причем широко распространены КМОП-аналоги БИС i8086 и i8088.

Для обеспечения интерактивного режима, при ведении баз данных и обработке графической информации необходимо высокое быстродействие портативной ЭВМ. В связи с этим (например, в модели IX-20 фирмы Epson) применяются два периферийных процессора, которые освобождают основной процессор от выполнения функций, связанных с вводом-выводом информации. Другие фирмы (рис. 2) для повышения производительности своих портативных персональных ЭВМ применяют сопроцессоры.

Судя по структурным схемам МПК последних выпусков, видно, что тенденция к увеличению числа микропроцессоров различного назначения в одной портативной ЭВМ устойчива.

Подавляющее большинство интегральных схем для МПК выполняются по КМОП-технологии. Ряд фирм для построения контроллеров внешних устройств применяют вентиляемые КМОП-матрицы.

Объем ОЗУ — 16...32К байт для «Lan-Size» МПК и 128...256К байт для «Full Function». Объем ПЗУ для типичных СПК — 32...200К байт.

Объем ОЗУ и ПЗУ наращивается подсоединением дополнительных микросхем или модулей памяти. Например, модель PX-8 фирмы Epson имеет 2 порта для подключения дополнительного модуля (по 32К байт), а дополнительные модули памяти модели HP-PLUS наращивают внутреннюю память до 3М байт. Отметим, что в некоторых ПМПК применяется ОЗУ с автономным источником питания. Это защищает занесенную в него информацию при отключении источника питания (HP-110 фирмы Hewlett-Packard).

Один из наиболее сложных узлов ПМПК — ЖК-экран. Использование высоких (вплоть до 100) коэффициентов мультиплексирования приводит к необходимости применения в экране совершенных ЖК-материалов. Даже при этом не удается получить больших углов обзора и достичь контрастности изображения выше 3:1 (все изготовители применяют ЖК-экраны с размерами точки не менее 0,5×0,5 мм). Для ПМПК типа «Lowend» наиболее характерны ЖК-экраны с емкостью не более 640 знакомест (16×40 или 8×80 символов).

В большинстве моделей с индикатором с повышенным коэффициентом мультиплексирования имеются ручные регуляторы напряжения возбуждения. Эти регуляторы позволяют подобрать оптимальные в конкрет-

ных условиях характеристики экрана по яркости (если можно говорить о такой характеристике для пассивного индикатора) и контрастности.

ЖК-экраны для ПМПК группы «Fullsize» имеют объем отображаемой информации значительно больший, чем у ПМПК группы «Lowend», и используют стандартное число знаков в строке — 80, а число строк — от 16 (HP-110) до 25 (11—11c, Data General/ONE); в графическом режиме — от 480×128 до 640×200 точек.

Для повышения наглядности информации широко используется программное изменение размеров символов экрана. Так, ЖК-экран модели PX-8 фирмы Epson позволяет работать с символами из 5×7 и 6×8 точек.

При разработке ПМПК фирмы-изготовители стараются достичь их максимальной аппаратной и программной совместимости с настольными (DESKTOP) ПЭВМ, которые используют монохроматические полноформатные мониторы, стандартные ПО и периферийное оборудование. Для этого в большинстве ПМПК использованы наиболее популярные последовательные (RS-232, RS-232C), параллельный (CENTRONICS) и специализированные интерфейсы (наиболее применимые в серии моделей, выпускаемых одной фирмой).

Среди интерфейсов МПК широко распространены касетный, видео. Например, контроллер Disk/Video фирмы Radio Shark содержит интерфейс подключения монохроматического монитора и обычного НГМД. Фирмы-разработчики МПК стремятся расширить рабочую область памяти, разрабатывая интерфейсы для подключения как внешних НГМД, так и накопителей типа «Винчестер» (Apple IIc).

Обеспечить передачу данных между ПМПК и другими ЭВМ важно уже при проектировании ЭВМ данного типа. Почти все ПМПК имеют для этого телефонные модемы. Ранние ПМПК имеют встроенные акустические модемы (200...300 Бод). Для увеличения скорости обмена информацией ЭВМ группы «Full Function» с акустическими модемами могут подключаться к внешним модемам (1200 Бод). Например, модем 8401 А фирмы NEC и модем МПК D.G./ONE фирмы Data General.

Клавиатура располагается, как правило, в корпусе ПМПК. В зависимости от размеров ПМПК используется 56...83 буквенно-цифровых клавиш со стандартным расположением пишущей машинки (QWERTY) и шагом 19,5 мм. Для облегчения работы в некоторых типах ПМПК (Ariscot, Stim PC) предусмотрены дополнительные цифровые, курсорные, функциональные и калькуляторные клавиши, выделяемые цветом. Имеются также модели (например, Ariscot), клавиатура которых соединяется с ЭВМ оптическим кабелем или инфракрасным каналом.

Используемые совместно с ПМПК периферийные устройства (встраиваемые или внешние) чрезвычайно разнообразны. Основные тенденции их развития — снижение массы и потребляемой мощности. Наиболее характерное встраиваемое периферийное оборудование — НГМД (133 и 82,5 мм) и микрокассетные магнитофоны. Отметим стремление обеспечить программно-аппаратную совместимость более компактных и экономичных дисководов 82,5 мм с дисководами 133 мм.

Печатающие устройства — обычно внешние. Подключение графопостроителей расширяет возможность ПМПК по выводу графической информации. Для автономной работы питание периферийного оборудования — в основном батарейное.

Учитывая то, что ЭВМ обладают достаточно высокой производительностью и широким ассортиментом используемого периферийного оборудования, фирмы-изготовители МПК предусмотрели возможность их работы и от встроенных источников тока, и от сети переменного тока общего пользования. Как правило,

встроенные источники тока выполнены на основе никель — кадмия или лития и допускают перезарядку. Во многих ПМПК средства контроля сигнализируют о необходимости перезарядки внутренних источников тока.

Программное обеспечение ПМПК имеет две устойчивые тенденции.

В моделях типа «Lowend» используются всего лишь специализированные (ориентированные на конкретные аппаратные возможности) программы обработки данных и управления, зашиваемые в ПЗУ. Только немногие устройства этого типа имеют возможность (при подключении к ним внешних НГМД) работать со стандартными ОС. Каждая такая ЭВМ имеет 3...7 ориентированных на неквалифицированного пользователя программ, обеспечивающих основные применения упрощенных систем (Wordstar, Spreadsheet, Personal Filer, Calendar, Multiplan).

В ЭВМ группы «Full Function» всегда предусматривается возможность работы со стандартными ОС (например, MS-DOS фирмы Microsoft). В этом случае монитор ОС, текстовые программы, ряд наиболее употребляемых утилит хранятся (для уменьшения времени работы НГМД) в ПЗУ. Использование стандартной ОС открывает перед пользователем такой ЭВМ практически неограниченные возможности использования ПО, наработанного для настольных ПЭВМ.

Конструкция ПМПК достаточно проста. ЭВМ группы «Lowend» — это моноблочные приборы, узлы электроники собраны на 1—3 двусторонних печатных платах.

Подавляющее большинство деталей отлиты из пластмасс. Конструктивы ЭВМ группы «Full Function» более сложны. Широко распространена конструкция с откидным экраном, позволяющая установить оптимальный угол наблюдения. Это немаловажно для применения ЖК-экранов с высоким коэффициентом мультиплексирования. В ряде случаев применены встроенные экономичные и миниатюрные НГМД и печатающие устройства.

Перспективы развития ПЭВМ с ЖК-экраном

Внимание зарубежных специалистов обращено сегодня к наиболее мощным ПМПК, совместимым с ПЭВМ фирмы IBM.

У ПЭВМ класса «Briefcase» в ближайшие годы при сохранении размеров, массы, клавиатуры и стоимости будут существенно улучшаться функциональные характеристики. Исходя из динамики роста возможностей технологии КМОП БИС, можно ожидать увеличения объема ОЗУ (вдвое) и быстродействия ЭВМ (на 50 % каждые 2—3 года). Типичные ПМПК 1990 г. будут иметь объем внутреннего ОЗУ в районе 2...4 М байт и быстродействие ЦП порядка 2...5 млн. операций/с. В этот же период, видимо, большинство ПЭВМ, в том числе МПК, будут переведены на 32-разрядную архитектуру. Ставится задача — достичь высокой контрастности изображения и снизить уровень потребляемой мощности. Только существенное повышение контрастности позволит полноценно использовать МПК для инженерных применений.

Ряд фирм разрабатывает цветные ЖК-экраны для перспективных МПК и экономичные устройства речевого ввода-вывода для портативных ПЭВМ.

Малые габариты РМПК не позволяют применить в них удобные клавиатуры и ЖК-экраны (круг пользователей таких ЭВМ ограничивается учащимися, отдельными энтузиастами). Видимо, РМПК будут все более специализировать для конкретных узких областей применения. За счет специализации можно значительно улучшить их эргономические характеристики (уменьшить число клавиш, увеличить ЖК-дисплей и т. д.). Важную роль малогабаритные ЭВМ будут играть в телекоммуникации: уже появляются высокоперспективные миниатюрные связанные терминалы на базе РМПК.

В ближайшее время можно ожидать разработку

РМПК с высокими технологическими характеристиками, включая большой ЖК-экран и 16-разрядный процессор.

Развитие относительно новых видов и способов обмена информации между человеком и ЭВМ позволяет надеяться на создание миниатюрных, удобных в обращении безклавиатурных ПЭВМ с большим (во всю поверхность прибора) индикатором, который одновременно может быть и средством для ввода информации. ЭВМ такого типа могут появиться на западном рынке в 1988—1990 гг.

Таким образом, развитие портативных ЭВМ обоих классов представляет собой важную проблему информатики и вычислительной техники и требует самого пристального внимания.

Телефон для справок: 460-14-56, Москва.

ЛИТЕРАТУРА

1. Computer-Age Mini-Micro Computer.—1985.— № 1.— P. 3.
2. Mini-Micro Systems.—1985.— № 3.— P. 50.
3. Computers.—1985.— № 2.— P. 126.
4. Computer-Age Mini-Micro Computer.—1985.— № 8.— P. 73.
- 5—12. Byte.—1985.— № 3.— P. 40; № 10.— P. 270, 443; № 2.— P. 39, 43; № 2.— P. 302; № 6.— P. 202; № 5.— P. 276.
- 13—15. Popular Electronics.—1981.— № 9.— P. 10; № 10.—104; № 11.— P. 35.
16. Byte.—1985.— № 7.— P. 129.
17. Popular Electronics.—1983.— № 11.— P. 49.
18. Computer Design.—1983.— № 12.— P. 54.
19. Popular Electronics.—1982.— № 2.— P. 84.
- 20—22. Byte.—1985.— № 1.— P. 283, 283; № 1.— P. 290, 429.
- 23—26. Popular Computing.—1984.— № 1.— P. 91; № 2.— P. 38; 1985.— № 5.— P. 54; 1984.— № 5.— P. 54.
- 27—28. Office Equipment News.—1985.— № 10.— P. 71.
29. Busines System.—1985.— № 9.— P. 6.
30. What Micro.—1985.— № 11.— P. 124.
- 31—32. Computer Systems.—1985.— № 9.— P. 84, 178.
33. Popular Electronics.—1980.— № 12.— P. 29.
- 34—35. What's news in Computing.—1984.— № 11.— P. 51; № 10.— P. 94.
36. Interface.—1985.— № 5.— P. 40.
37. Computer Personlich.—1985.— № 8.— P. 8.
38. Electronique Actualities.—1984.— № 10.— P. 8.
- 39—41. Computer Weekly.—1985.— № 14. Febr.— P. 24; № 21.— Febr.— P. 33; 1984.— № 13.— Sept.— P. 34.
- 42—51. Byte.—1984.— № 4.— P. 229; № 5.— P. 276; № 6.— P. 52, 110; № 7.— P. 339, 102; № 11.— P. 416; № 3.— P. 346; № 10.— P. 42, 63.
- 52—53. Portable.—1985.— № 3.— P. 12, 73.
- 54—58. Micro Personal Computer.—1985.— № 9.— P. 21, 64.
59. Ordi Magazin.—1985.— № 13.— Febr.— P. 63.
60. What's News in Computing.—1985.— № 7.— P. 31, 64, 77, 70.
61. Byte.—1985.— № 1.— P. 39.
- 62—65. Popular Computing.—1984.— № 6.— P. 96, 36; № 9.— P. 134; № 6.— P. 50.
66. Busines Computing.—1985.— № 9.— P. 21.
- 67—69. Popular Computing.—1985.— № 7.— P. 82; № 6.— P. 101; № 9.— P. 20.
70. Popular Electronics.—1984.— № 9.— P. 34.
71. Computer.—1985.— № 21.— Febr.— P. 126.
- 72—73. Computer.—1985.— № 1.— P. 301; № 2.— P. 41.
- 74—76. Mini-Micro System.—1984.— № 10.— P. 54, 54, 217.
77. Computer Weekly.—1984.— № 4.— July.— P. 39.
- 78—83. Computer Design.—1982.— № 5.— P. 68, 196; 1981.— № 4.— P. 262; 1982.— № 3.— P. 220, 236; № 4.— P. 222.
- 84—85. Computers.—1984.— № 5.— P. 96, 88.
- 86—87. Computer Design.—1983.— № 12.— P. 159; № 9.— P. 291.
- 88—99. Popular Computing.—1985.— № 2.— P. 117; № 3.— P. 110; № 6.— P. 82; № 5.— P. 11, 82, 86, 139; № 6.— P. 17, 32.
- 100—102. Computer.—1984.— № 2.— P. 94, 90; № 3.— P. 91.
- 103—106. Byte.—1983.— № 7.— P. 519; 1982.— № 9.— P. 166; 1983.— № 11.— P. 56, 373.
107. Computer World.—1985.— Vol. 39.— № 9.— P. 1.
108. Datamation.—1983.— Vol. 19.— № 11.— P. 173.
- 109—110. Byte.—1983.— № 3.— P. 85, 183.
111. Popular Computing.—1984.— № 5.— P. 72.
112. Computers.—1985.— № 2.— P. 124.
113. Computer Design.—1985.— № 7.— P. 167.
114. Mini-Micro Systems.—1984.— № 10.— P. 233.
115. Byte.—1985.— № 7.— P. 252.
116. Practica Computing.—1985.— № 7.— P. 62.
- 117—118. What's Micro.—1985.— № 11.— P. 137.
119. Interfase.—1985.— № 2.— P. 13.
- 120—122. Computer Products.—1982.— № 4.— P. 49; № 2.— P. 16, 20.
- 123—127. Creativ Computing.—1983.— Vol. 9.— № 12.— P. 20; № 8.— P. 37, 190, 194, 194.
- 128—129. Electronics Today.—1985.— № 8.— P. 86, 91.
130. Electronics Industry.—1985.— № 9.— P. 57.
131. Creative Computing.—1984.— Vol. 10.— № 10.— P. 176.
- 132—133. Personal Computing Technology.—1985.— Vol. 3.— № 9.— P. 164, 165.
134. Popular Science.—1983.— № 7.— P. 60.
135. Computers.—1984.— № 5.— P. 90.
136. Byte.—1983.— № 4.— P. 124.
137. Computer Personlich.—1984.— № 16.— Oct.— P. 136.
138. Electronics News.—1985.— № 22.— Apr.— P. 34.
139. Computer Design.—1984.— № 6.— P. 128.
140. What's News in Computing.—1985.— Aug./Sept.— P. 44.
141. Popular Science.—1985.— № 5.— P. 75.
142. Byte.—1986.— № 11.— P. 73.
143. Byte.—1986.— № 9.— P. 38.

Уважаемая редакция!

С удивлением читал я статью [1]. Оказывается, я автор и апологет прозатратного метода оценки труда программистов!

Но, позвольте!

Во-первых, авторство — слишком большая честь для меня: я сам родился позже указанного метода.

Во-вторых, апологетом метода я также не был. Более того, статья [2] была посвящена как раз критике этого метода.

Вот тезисы, которые я отстаиваю на самом деле:

1. Оценивать производительность труда программистов при помощи длины или любой другой характеристики программы в корне неправильно.

2. Программирование (по крайней мере, профессиональное программирование) — это промышленная деятельность, однако из этого вовсе не следует, что программа — это изделие. Прежде, чем переносить промышленные методы (в частности, методы оценки производительности труда) в программирование, следует провести прагматический анализ программирования как деятельности и корректно определить понятия «изделие», «инструмент» и т. д.

3. По своей природе программа ближе к технологии, чем к изделию, и именно из этого нужно исходить, оценивая деятельность программистов.

(Продолжение см. на с. 57)

МАШИННАЯ ГРАФИКА

УДК 681.03.06

Ю. Л. Кетков

ГРАФИЧЕСКИЕ СРЕДСТВА В АЛГОРИТМИЧЕСКОМ ЯЗЫКЕ БЕЙСИК

Элементы машинной графики появились в языке Бейсик сравнительно недавно. Первый стандарт ANSI, принятый в 1978 г., их не предусматривал. Однако бурный рост выпуска ПЭВМ и развитие индустрии «машинных игр» в значительной мере содействовали появлению новых версий языка, снабженных средствами обработки и отображения графической информации.

В настоящей статье достаточно подробно рассматриваются графические компоненты в четырех реализациях языка Бейсик на персональных ЭВМ «Искра 226», IBM PC, MSX и «Электроника 85». Выбор указанных версий позволяет продемонстрировать различные аспекты программного обеспечения машинной графики — многоуровневую систему графических процедур («Искра 226»), основные элементы цветной дисплейной графики (IBM PC, MSX), влияние идей проекта нового стандарта Бейсика на графическую компоненту языка («Электроника 85»).

Машинная графика на ПЭВМ «Искра 226»

В качестве технических средств отображения графической информации на ПЭВМ «Искра 226» используются алфавитно-цифровой дисплей с графическими возможностями и малогабаритный графопостроитель Н-306. В текстовом режиме дисплей воспроизводит содержимое собственного буферного ЗУ емкостью 1920 байт (24 строки по 80 символов). В графическом режиме дисплейный кадр представлен растром 560×256 (в более поздних моделях 512×256) точек, на котором отображается специальным образом подготовленная информация в буфере, выделяемом программой пользователя. Началу координат соответствует нижний левый угол экрана. Разрешающая способность дисплея в графическом режиме составляет 0,4 мм. Выводимый текст и графическое изображение на экране совмещены. Графопостроитель Н-306 имеет рабочее поле размером 300×200 мм, соответствующее растру 4000×2666 точек. Разрешающая способность графопостроителя равна 0,075 мм.

Средства языка Бейсик-Искра предусматривают три уровня взаимодействия с устройствами отображения графической информации.

На первом уровне используются элементарные графические команды, обеспечивающие заданные относительные перемещения пишущего узла. Графическая команда состоит из однобайтового кода операции, задающего характер (направление) движения, и одного-двух операндов, определяющих величины смещений по осям x и y . Шестнадцатеричные коды команд движения луча (пера) приведены на рисунке. Некоторые элементарные графические команды задаются только кодом операции и в дополнительной информации не нуждаются. К ним относятся процедуры включения или отключения засветки (для графопостроителя — опускания или подъема пера), имеющие соответственно коды 06 и 07, возврата графического курсора (01), очистки экрана (0D), блокировки (0E) или отмены блокировки (0F) отображения графического объекта на экране.

Исполнение элементарной графической команды осуществляется оператором PRINT. Например:

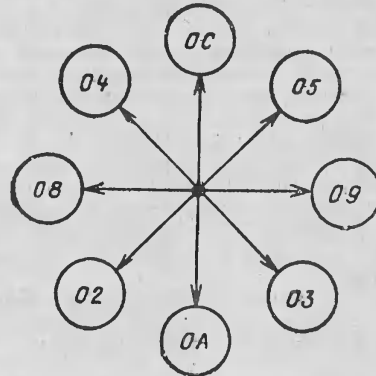
150 PRINT / 10 HEX (05), 100, 200

250 PRINT / 14 HEX (05), 100, 200

Элементарные графические команды позволяют воспроизводить на графопостроителе и отдельные символы. Например:

105 PRINT / 14 HEX (E6E 5F4EFF720E02EEC2E)

Каждая пара шестнадцатеричных цифр представляет здесь графическую команду по воспроизведению отдельного символа размером 12×16 дискретов и межсимвольным расстоянием в 7 дискретов. Полный список кодов графических команд можно найти в работе [1].



Коды команд, определяющие направление движения

Таблица 1

Перечень графических макрокоманд

Формат макрокоманды	Выполняемая функция
<„ R>	Быстрая установка пишущего узла в начало координат (левый нижний угол)
<ΔX, ΔY, U>	Перемещение пишущего узла из текущей точки (X_T, Y_T) в точку ($X_T + \Delta X, Y_T + \Delta Y$) без фиксации следа пера
<ΔX, ΔY, D>	Аналогичное перемещение с вычерчиванием отрезка прямой
<n, C>	Установка масштаба (n) вычерчивания символов (стандартный размер символа 5×7 дискретов)
<ΔX, ΔY, S>	Установка горизонтального (ΔX) и вертикального (ΔY) смещений между соседними символами. По умолчанию ΔX=2, ΔY=0
<ΔX, ΔY „S ₁ S ₂ ...S _K “>	Воспроизведение текстовой константы, заданной в кавычках. Значения ΔX и ΔY определяют соответствующие смещения нижнего левого угла знакоместа первого символа от текущей точки

Второй уровень процедур отображения графической информации связан с использованием логического устройства PLOT и оператора с одноименным идентификатором. Работе с оператором PLOT предшествует назначение графопостроителя или графического дисплея логическому устройству и выделение ему буфера в оперативной памяти:

200 SELECT PLOT 14 (4000)

Операндами в теле оператора PLOT являются графические макрокоманды, состоящие из тройки параметров, заключенных в угловые скобки $\langle \alpha, \beta, \gamma \rangle$. Один оператор PLOT может содержать несколько таких макрокоманд. Полный перечень графических макрокоманд второго уровня приведен в табл. 1. Относительные перемещения $(\Delta X, \Delta Y)$ в графических макрокомандах могут задаваться арифметическими выражениями.

Наибольшие возможности по отображению геометрической информации и конструированию описаний графических объектов (ГО) предоставляют операторы третьего уровня.

Графический объект, который можно программным путем создавать, преобразовывать и выводить, представляет собой совокупность точек, векторов и надписей. Формируется ГО обычно в декартовой системе координат ($0 \leq x \leq 32767, 0 \leq y \leq 32767$), единицей длины в которой служит дискрет экрана или графопостроителя.

Для хранения информации, описывающей структуру ГО и установочные данные, необходимые для его размещения на рабочем поле, применяются одномерные или двумерные текстовые (символьные) массивы. Имя такого массива отождествляется с именем ГО. По сути дела, текстовый массив используется в данном случае лишь как средство резервирования памяти (буфера) под описание ГО. Размер области памяти, запрашиваемый с помощью оператора DIM, должен быть не менее 64 байт. По умолчанию, для хранения ГО выделяется 1600 байт.

Перед началом работы с любым ГО его надо предварительно активизировать (открыть), т. е. употребить имя ГО в теле оператора \square OPEN. Например:

100 \square OPEN A \square (), B \square ()

Имя ГО, указанного в списке оператора \square OPEN на первом месте, выделяет графический объект, именуемый экранным. Отображение этого объекта, если в его описании уже что-то есть, немедленно выдается на экран дисплея. Если с экраным ГО производятся какие-то операции (дополнение новых элементов, уничтожение старых, перенос начала координат, повороты и т. п.), то их результат можно сразу же увидеть на экране.

Если необходимо открыть дополнительные графические объекты C \square и D \square , но при этом сохранить объект A \square в качестве экранного, то можно воспользоваться модификацией оператора \square OPEN:

180 \square OPEN, C \square (), D \square ()

Процедура активизации графического объекта с помощью оператора \square OPEN заключается не только в том, что мы получим возможность доступа к этому ГО. Для пустого, т. е. еще не сформированного ГО, открытие сопровождается начальным формированием установочных данных. К ним относятся координаты точки привязки объекта, определяющие его расположение на рабочем поле, масштабные коэффициенты по осям x и y, угол поворота объекта относительно физической системы координат, минимальное и максимальное значения горизонтальной и вертикальной координат, используемые при проектировании ГО на экран дисплея, координаты текущей точки.

Стандартные установочные данные могут быть изменены пользователем для того, чтобы переместить ГО или повернуть его на заданный угол, произвести масштабирование по одной или обеим координатам и т. п.

В описании ГО важную роль играет понятие текущей точки. Пока объект пуст, текущая точка совпадает с началом локальной системы координат. По мере пополнения объекта точками, векторами и надписями каждая последняя точка очередной компоненты становится текущей. Если спустя какое-то время мы продолжим формирование ГО, то новые фрагменты начнут пристраиваться к заполненной текущей точки. При построении многосвязных фигур положение текущей точки в описании ГО можно изменить с помощью оператора NPLOT или DNPlot.

Включение новой точки в описание графического объекта осуществляется с помощью оператора DOT или DDOT:

150 DOT G \square (), X, Y

300 DDOT G \square (), -20, 40

В первом случае точка задается координатами (X, Y) в локальной системе координат объекта, во втором — перемещениями $(\Delta X, \Delta Y)$ относительно текущей точки.

Вектор из текущей точки в точку, заданную локальными координатами (X, Y) или относительными перемещениями $(\Delta X, \Delta Y)$, добавляется к описанию ГО с помощью операторов DRAW или DDRAW.

Для включения в состав описания ГО пояснительной подписи, заданной значением текстовой константы или текстовой переменной, используется оператор LABEL:

220 LABEL A \square (), k, $\Delta_x, \Delta_y, "S_1 S_2 \dots S_n"$

Параметры k, Δ_x, Δ_y задают соответственно масштаб изображения символов (по умолчанию, k=1, стандартный размер знакоместа 7x5 дискретов), горизонтальное (Δ_x) и вертикальное (Δ_y) смещения между двумя соседними символами (по умолчанию, $\Delta_x=2, \Delta_y=0$). Надпись включается в конфигурацию ГО таким образом, что левый нижний угол знакоместа первого символа совпадает с положением текущей точки.

Элементарные графические фигуры — видимые (DOT, DDOT) или невидимые (NPLOT, DNPlot) точки, векторы (DRAW, DDRAW) и надписи (LABEL), входящие в состав ГО, нумеруются в порядке их подключения. Это обеспечивает возможность доступа к любой компоненте графического объекта с целью ее модификации, исключения или использования при компоновке других ГО. Все эти функции выполняются с помощью специального оператора присваивания:

100 \square LET A \square (5) = 0

200 \square LET A \square () = A \square () + B \square (2)

300 \square LET A \square (4) = B \square (8)

В правой части оператора LET может находиться любое число «слагаемых», каждое из которых представлено графическим объектом или любой компонентой ГО. Порядок слагаемых определяет последовательность их воспроизведения и соответственно нумерацию вновь присоединяемых компонент.

Удаление точек и векторов из описания ГО можно выполнить и с помощью операторов NDOT и NDRAW. Если удаляемая компонента была построена с помощью абсолютной адресации (DOT, DRAW), то в теле соответствующего оператора надо будет задать те же координаты точки или конца вектора. Если же объект был включен в ГО с использованием относительных перемещений, то для его удаления придется вычислить абсолютные координаты.

Описание графического объекта может быть подвергнуто одному или любой последовательности аффинных преобразований — сдвигу (C MOVE), повороту (TURN) вокруг заданной точки и процедурам сжатия — растяжения (STRETCH) относительно указанной точки. Все эти операторы не сопровождаются пересчетом координат каждой точки в описании ГО, а приводят только к изменению результирующей матрицы преобразований. И лишь для экранного объекта действительные результирую-

шей матрицы распространяется на каждую точку описания.

Сформированное описание графического объекта может быть сохранено в архиве (DATA SAVE). В дальнейшем его можно оттуда извлечь (DATA LOAD) и использовать для получения графической копии или формирования более сложных ГО.

Выводу графического объекта на экран дисплея или на графопостроитель могут предшествовать процедуры, согласующие размеры области вывода (WINDOW) и предельные границы изменения координат объекта (FRAME). С помощью оператора WINDOW задаются размеры прямоугольной области (X_{min} , X_{max} , Y_{min} , Y_{max}), которая должна целиком помещаться на рабочем поле дисплея или графопостроителя. Отрезки прямых (векторы), выходящие за пределы области вывода, отсекаются по границе области. Если на эту границу попадает текст, то он не выводится. Оператор FRAME, имеющий аналогичные параметры, задает размеры прямоугольной области, в которой целиком помещается ГО. Если координатная сетка не совпадает с растром экрана или графопостроителя, то указанные пределы используются для преобразования программных координат в физические координаты области вывода.

Среди дополнительных возможностей по согласованию программных и физических областей отметим операторы ORIGIN и SCALE, обеспечивающие соответственно перенос начала координат ГО и изменение цен делений по осям x и y . Графический объект, объявленный экранным, выдается на дисплей автоматически. Для вывода этого объекта или любого другого открытого ГО на графопостроитель предназначен оператор \square COPY:

```
120  $\square$  COPY/14, A  $\square$  ( )
```

В качестве дополнительных параметров в этом операторе могут быть указаны координаты нижнего левого угла окна вывода. Графический объект, не объявленный экранным, но открытый по оператору \square OPEN, может быть выведен и на дисплей:

```
130  $\square$  COPY/10, B  $\square$  ( ), 200, 100
```

Дополнительные средства по редактированию графического объекта предоставляет световое перо.

Цветная дисплейная графика на IBM PC

IBM PC [2] может быть укомплектована цветным монитором, который работает в алфавитно-пиксельном или графическом режиме. Переключение с одного режима на другой осуществляется программным путем по оператору SCREEN n ($n=0, 1, 2$).

В графическом режиме дисплей допускает две степени разрешения — среднюю ($n=1$) с растром 200×320 и высокую ($n=2$) с растром 200×640 . Точки графического раstra нумеруются, начиная с нижнего левого угла. По вертикали эти номера идут от 0 до 319 (639), по горизонтали — от 0 до 199. В среднем разрешении каждая точка раstra может быть окрашена в один из четырех цветов, представленных в выбранной пользователем палитре. Для точек высокого разрешения цветовая гамма состоит всего из двух красок.

Буферное запоминающее устройство, в котором хранится отображаемая графическая информация, имеет емкость 16К байт. Оно позволяет разместить сведения о 64000 пикселей при среднем разрешении или о 128000 пикселей при высоком разрешении.

Графические операторы входного языка BASIC-A в явном виде не разделяются на трехуровневую схему, описанную выше. Уровень физического управления движением луча, а также средства описания и воспроизведения графических объектов здесь совмещены в операторе DRAW:

```
DRAW {  $\begin{matrix} v \square \\ "S_1 S_2 \dots S_k" \end{matrix}$ 
```

Фрагмент графической программы, записанный в теле

оператора DRAW с помощью значения текстовой переменной или литеральной константы, представляет собой набор закодированных элементарных команд.

Каждая команда содержит код операции (один или два символа) и соответствующий набор данных. Допускается два формата записи графических команд: без разделителя (например, R5) или в форме оператора присваивания (например, R=5). В качестве второго символа кода операции может использоваться буква V, которая в сочетании с одной из девяти команд перемещения блокирует появление на экране соответствующего отрезка, т. е. движение осуществляется без заставки луча.

Выполнение команды X можно рассматривать как обращение к графической подпрограмме. Эта дает нам возможность формировать фрагменты ГО, присваивать им имена и в дальнейшем использовать их при компоновке или прорисовке более сложных изображений. Однако указанная процедура обладает гораздо меньшими возможностями по сравнению со средствами описания и манипуляции с ГО в языке Бейсик-Искра.

К числу процедур, используемых для формирования на экране дисплея графических примитивов, относятся операторы воспроизведения точек основным (PSET) или фоновым (PRESET) цветом, линий или прямоугольников (LINE), дуг окружностей или эллипсов (CIRCLE).

Любая ограниченная область, построенная тем или иным способом на экране дисплея, может быть окрашена с помощью оператора PAINT:

```
PAINT (X, Y) [, C0 [, Cr]]
```

Здесь (X, Y) — абсолютные или относительные координаты любой внутренней точки окрашиваемой области. Код цвета C_0 определяет оттенок, приписываемый всем внутренним точкам указанной области. Код цвета C_r присваивается граничным точкам области.

Графические объекты, которые мы видим на экране, представляют собой отображение буферной памяти дисплея. Каждому пикселю экрана IBM PC соответствует два бита в буферном ЗУ при среднем разрешении или один бит — при высоком. В этих разрядах хранится код цвета соответствующей точки экрана. Если вырезать на экране некоторую прямоугольную область, то соответствующая информация о состоянии всех ее точек может быть развернута в двоичный вектор. Содержимое этого вектора или информацию об отдельной его компоненте можно прочитать из видеоЗУ в оперативную память ПЭВМ или произвести обратную операцию. Для выполнения этих действий используются функция POINT, операторы GET и PUT.

С помощью функции POINT можно прочитать состояние отдельной точки дисплея, расположенной на пересечении i -й строки и j -го столбца экранного раstra. Например:

```
100 N = POINT (140, 255)
```

Таким образом можно прочитать в цикле и запомнить состояние всех точек некоторой прямоугольной области экрана. Однако это будет крайне неэффективно как по скорости, так и по объему использованной оперативной памяти. Существенный выигрыш по обоим параметрам достигается с помощью оператора GET. Например:

```
120 GET (10, 20) — (60, 80), A
```

Первый параметр задает границы считываемого раstra с помощью координат двух противоположных вершин прямоугольника. Второй параметр определяет имя массива достаточной длины, на место которого считывается содержимое выделенного фрагмента видеоЗУ.

Считанную информацию можно подвергнуть программной обработке и вновь выдать на экран с помощью оператора PUT.

Техника блуждающей маски на ПЭВМ типа MSX

ПЭВМ типа MSX (MSX-2) принадлежит к числу наиболее распространенных домашних компьютеров, выпускаемых целым рядом фирм Японии. Своим названием

эти компьютеры обязаны торговой марке фирмы Microsoft, разработавшей для указанных ПЭВМ операционную систему MS-DOS и целую серию интерпретаторов языка Бейсик.

Графическая компонента входного языка BASIC-MSX сравнительно мало отличается от описанных выше процедур для IBM PC. Небольшие отклонения вызваны разрешающей способностью экрана (192×256 точек в высоком разрешении и 48×84 — в низком), другим расположением координатных осей (ось y направлена вниз, а начало координат расположено в верхнем левом углу), более богатой цветовой гаммой.

Однако самое существенное отличие MSX от IBM PC заключается в использовании 32 электронно-управляемых масок, расположенных перед главным экраном. На главном экране формируется основное графическое изображение. Здесь же располагается и текстовая информация. Сквозь незанятые позиции главного экрана «просвечивают» фоновый экран и окаймляющая его цветная рамка.

Смысл экранных масок состоит в том, что нанесенные на них разноцветные фигурки или узоры накладываются на основное изображение, затеняя и дополняя его. Более того, благодаря перемещению отдельных фигурок (узоров) по площади маски можно оживить общее изображение. При этом детали, изображенные на масках с меньшими номерами, заслоняют все то, что находится дальше от пользователя.

На одной маске можно расположить единственное одноцветное изображение, напоминающее мозаику на поле 8×8 или 16×16 квадратиков. Размеры этих квадратиков либо совпадают с размерами точек на главном экране (масштаб 1:1), либо превышают их по площади в 4 раза (масштаб 2:1).

Для описания изображения используется набор символьных констант, значения которых представлены битовыми шкалами. Шаблон 8×8 описывается с помощью восьми однобайтовых констант, и каждая из них соответствует одной строке изображения. Обычно строка задается двумя шестнадцатеричными цифрами, двоичное представление которых однозначно определяется расположением светлых (нули) и темных (единицы) пятен в клеточках строки. Символьные переменные — строки объединяются (операция конкатенации) в общую текстовую константу, значение которой присваивается одной из переменных служебного массива `SPRITE`.

При кодировании изображения, расположенного на поле размером 16×16, шаблон разбивают на четыре части. Для каждой из этих частей формируется своя текстовая константа, а их «сумма» заносится на место выбранного элемента массива `SPRITE`.

Всего в служебном массиве `SPRITE` можно запомнить до 256 различных описаний шаблонов размера 8×8 или до 64 описаний шаблонов размера 16×16. Выбор размера шаблона и масштаба его клеточек управляется вторым параметром оператора `SCREEN`:

```
SCREEN 2[, j]
```

Для переноса закодированного изображения из k -го элемента массива `SPRITE` на маску с номером i используется оператор `PUT SPRITE`:

```
PUT SPRITE i, (X, Y), C, k
```

Здесь (X, Y) — координата точки привязки шаблона (верхний левый угол) на поле маски, C — код цвета темных пятен в шаблоне ($0 \leq C \leq 15$).

Выполняя процедуру `PUT SPRITE` в цикле и меняя каждый раз координаты (X, Y) точки привязки, можно имитировать движение объекта по экрану. Если на двух или более масках расположены перемещающиеся объекты, то аппаратные средства позволяют определять момент, когда разные изображения начинают накладываться друг на друга. Эта ситуация вызывает прерывание, которое можно обработать программным путем (`ON`

`SPRITE GOSUB ...`, `SPRITE ON`, `SPRITE OFF`, `SPRITE STOP`).

Предложения по стандартизации графической компоненты

Проект нового стандарта языка Бейсик [3] включает и предложения по графике, однако они не очень удачно согласуются с уже принятым международным графическим стандартом GKS [4]. Поэтому излагаемый ниже материал можно рассматривать лишь как некоторое приближение к стандарту на взаимодействие с графическим дисплеем. Ориентация на дисплей не налагает запрета на использование других графических устройств. Однако при работе с пассивными регистраторами изображений пользователь может употреблять только операторы графического вывода.

Рабочее поле представляется максимальным квадратом, вписывающимся в отображаемую область экрана. На нем вводится декартова система координат с началом в нижнем левом углу. Вернем правому углу квадрата соответствует точка с координатами (1,1). Такую систему координат принято называть «мировой».

Если отображающая поверхность экрана представляется собой прямоугольник, вытянутый по горизонтали или вертикали, то рабочий квадрат прижимается к правому верхнему углу. Расположение рабочего поля на экране может быть изменено путем ограничения ширины и высоты части экрана, используемой под графику (`SET BOUNDS X, Y`).

Выделение прямоугольной области в пределах рабочего поля, предназначенной для отображения очередной порции графических данных, осуществляется с помощью оператора `SET VIEWPORT`:

```
SET VIEWPORT Xmin, Xmax, Ymin, Ymax
```

Указанная здесь четверка значений, каждое из которых должно принадлежать интервалу [0, 1], выделяет очередную область рисунка.

Программные координаты отображаемых точек в экранные координаты не переводятся автоматически. Для установки пределов изменения программных значений по координатам X и Y соответственно используется оператор `SET WINDOW`:

```
SET WINDOW Xmin, Xmax, Ymin, Ymax
```

Простейшими операторами, управляющими перемещением луча в выделенной области, являются `PLOT`, `MAT PLOT` и `MAT FILL`. Элементами выводного списка оператора `PLOT` служат пары арифметических выражений, значения которых воспринимаются как координаты X и Y очередной точки воспроизводимого полигона:

```
PLOT X1, Y1; X2, Y2; ...
```

Разделитель в конце списка означает, что последняя вершина, построенная данным оператором `PLOT`, будет соединена с первой точкой из следующего оператора `PLOT`. Для разрыва связи последней точки объекта, строившегося в цикле, используется пустой оператор `PLOT` (полная аналогия с оператором `PRINT`).

Оператор `MAT PLOT` оперирует двумя числовыми массивами, в которых должны быть сформированы координаты X и Y для всех вершин воспроизводимого полигона:

```
MAT PLOT A, B
```

Другая модификация оператора `MAT PLOT` позволяет работать с одним двумерным массивом C ($N, 2$), в котором координаты вершин попарно перемежаются: $C(1,1) = X_1$, $C(1,2) = Y_1$, $C(2,1) = X_2$, $C(2,2) = Y_2$, ... В этом случае формат оператора вывода таков:

```
MAT PLOT 50 : C
```

Число, расположенное перед именем массива, определяет число первых точек, участвующих в построении. Оператор `MAT FILL` отличается от оператора `MAT PLOT` тем, что соединяет последнюю точку воспроизводимого полигона с первой. В результате формируется

замкнутая многоугольная область, которая дополнительно может быть окрашена, если графический дисплей такую возможность предусматривает.

Существенно более широкие изобразительные средства обеспечиваются графическими процедурами, т. е. подпрограммами, формирующими изображение объекта с заданными параметрами. Описание таких объектов начинается с оператора PICTURE, в теле которого указываются имя ГО и список формальных параметров, заключенных в скобки. Описание ГО завершается оператором END PICTURE. Например, подпрограмма построения правильного n -угольника, вписанного в единичную окружность с центром в текущей точке, имеет следующий вид:

```
200 PICTURE POLYGON (N)
2010 LET P = 2*PI/N
2020 FOR I = 0 TO N
2030 PLOT COS (P*I), SIN (P*I)
2040 NEXT I
2050 PLOT
2060 END PICTURE
```

В описании одной графической процедуры может встретиться обращение к другой процедуре. Например, для воспроизведения окружности можно воспользоваться правильным многоугольником с достаточно большим ($n=100$) числом сторон. И тогда описание окружности может выглядеть следующим образом:

```
2100 PICTURE CIRCLE
2110 PLOT POLYGON (100)
2120 END PICTURE
```

Вызов и исполнение графической процедуры осуществляются оператором PLOT, расположенным среди группы выполняемых операторов. В отличие от этого оператор с меткой 2110 принадлежит к группе описательных операторов и немедленно исполнению не поддается.

В теле оператора PLOT кроме имени вызываемой процедуры и списка фактических параметров может содержаться ссылка на матрицу аффинных преобразований. Например:

```
100 PLOT POLYGON (4) WITH A
```

В этом случае координаты каждой воспроизводимой точки перед построением «умножаются» на матрицу преобразований, имеющую размерность 4×4 . Формирование матриц для наиболее распространенных геометрических преобразований осуществляется с помощью стандартных процедур, приведенных в табл. 2.

Каждая из этих процедур формирует свою матрицу преобразований. Однако их можно комбинировать, перемножая в нужном порядке для получения более слож-

ных результирующих матриц. Например:

```
100 MAT B = ROTATE (PI/4) * SCALE (2) *
    * SHIFT (2, 3, 3)
```

Оператор матричного присваивания в проекте стандарта специально расширен для формирования матрицы преобразований с помощью такого рода выражений.

Для совмещения графического изображения с текстовой информацией в проекте стандарта предусмотрен оператор GRAPHIC PRINT, обеспечивающий вывод пояснительных подписей в заданном месте экрана:

```
GRAPHIC PRINT AT X, Y: „S1S2...Sn“
```

Каждый символ отображаемого текста окаймлен квадратом, в котором условно введена локальная система координат с началом в левом нижнем углу знакоместа и точкой (1,1) в правом верхнем углу. Оператор GRAPHIC PRINT использует координату точки (X, Y) для привязки текста по следующему алгоритму:

если перед выводом был установлен «левый прижим» (SET JUSTIFY 0,0), то в точку с координатами (X, Y) помещается нижний левый угол знакоместа первого символа текста;

если перед выводом был установлен «средний прижим» (SET JUSTIFY 2.5, 2.5), то в точку (X, Y) помещается центр первого знакоместа;

если перед выводом был установлен «правый прижим» (SET JUSTIFY 1, 1), то в точку (X, Y) помещается правый верхний угол первого знакоместа.

Дополнительно к указанным средствам расположения надписи имеется возможность изменить высоту символов (SET CHARACTER HEIGHT) и установить угол наклона надписи (SET TEXT ANGLE).

Среди других графических возможностей, содержащихся в проекте стандарта, следует отметить операторы очистки экрана (CLEAR), установки основного (COLOR) и фонового (BACKGROUND COLOR) цветов, определения типа линии — сплошная, пунктирная или штрихпунктирная (SE LINE STYLE), включение или отключение режима отсечения частей изображения, выходящих за границы, установленные с помощью оператора VIEWPORT (SET CLIP ON, SET CLIP OFF).

Оператор ввода GRAPHIC INPUT позволяет локализовать позицию точки, в которой находится маркер (курсор) или какое-либо другое средство целеуказания (световое перо, рычажное или шаровое координатное устройство). Таким образом можно не только ввести координаты отмеченной точки, но и прочитать символ, расположенный в окрестности указанной точки.

К идеям графической компоненты стандарта наиболее близка версия Бейсик-Прос, функционирующая на ПЭВМ «Электроника 85».

Физическая система координат, привязанная к экрану дисплея, использует мировые координаты. Однако из-за вытянутости по горизонтали на рабочем поле размещается только часть «мирового» квадрата: $0 \leq x < 1, 0 \leq y < 0,625$.

По умолчанию, под графический экран отводится область, ограниченная линиями: $x_{min}=0,375, x_{max}=1, y_{min}=0, y_{max}=0,625$. Однако ее местоположение и размеры можно изменить с помощью оператора SET VIEWPORT.

С физической системой координат экрана связан графический маркер, положение которого соответствует текущей точке. В отличие от текстового курсора, положение которого отображается аппаратными средствами, графический курсор воспроизводится программным путем. Установка нового положения графического курсора производится с помощью оператора SET:

```
SET POSITION (ex, ey)
```

Для опроса координат текущей точки используется оператор ASK POSITION (X, Y).

Простейшие графические построения выполняются с помощью модификаций оператора PLOT:

```
PLOT (X1, Y1), ..., (Xn, Yn)
```

```
PLOT ARC (X0, Y0, φ)
```

```
PLOT CURVE (X(), Y(), N, I)
```

Таблица 2

Стандартные процедуры формирования матриц

Формат процедуры	Результат действия формируемой матрицы
SCALE (к)	Масштабирование, при котором точка (x, y) переходит в точку (к·x, к·y)
SCALE (к, р)	Раздельное масштабирование по каждой переменной: (x, y) → (к·x, р·y)
SHIFT (Δx, Δy)	Параллельный перенос: (x, y) → (x+Δx, y+Δy)
ROTATE (α)	Поворот на угол α, отсчитываемый от оси x против часовой стрелки
SHEAR (α)	Наклон вертикальных линий вправо на угол α

Перечень установочных процедур

Формат оператора	Выполняемое действие
SET CHARACTER S, M	Установка двоичной маски — матрицы M размером 8×8 или 16×16, используемой в качестве микрорастра при воспроизведении символа S
SET CHARACTER SIZE W, H	Установка ширины (W) и высоты (H) символов в дискретах экрана
SET CHARACTER SPACING ΔX, ΔY SET ITALICS φ	Установка межсимвольного зазора Установка угла наклона символов в градусах (при φ > 0 — левый курсив, при φ < 0 — правый)
SET TEXT ANGLE π	Установка одного из 8 (0 ≤ π ≤ 7) фиксированных углов наклона строки (0°, 51°, 90°, 129°, 180°, 231°, 270°, 309°)

Координаты точек в операторах PLOT и SET POSITION вовсе не обязательно задавать в системе физических координат. Для автоматического приведения программных координат к экранным необходимо предварительно объявить пределы изменения математических координат с помощью оператора SET WINDOW.

Если отдельные части воспроизводимой фигуры выйдут за пределы отведенной области, то можно продолжить построение за пределами графического экрана или отсечь выступающие детали. Режим отсекается оператором SET CLIP ON, а отключается оператором SET CLIP OFF.

Наряду с выводом простейших геометрических фигур в поле графического экрана можно отобразить текстовую и числовую информацию с достаточно широкими возможностями по управлению конфигурацией и расположением символов. Для этой цели используются операторы GRAPHIC PRINT и GRAPHIC PRINT USING. Списки вывода и средства описания формата числовых и символьных полей здесь те же, что и в обычном операторе PRINT. Основное же отличие заключается в том, что «графическая» печать учитывает такие параметры, как размер и начертание символов, межсимвольный зазор, ориентацию символов и угол наклона выводимой подписи (табл. 3).

Графический вывод может использовать один из трех шрифтов, каждый из которых включает 95 символов. Шрифт с номером 0 соответствует стандартному начертанию символов ASCII, маски микрорастров которых заложены в ПЗУ. Два других шрифта с номерами 1 и 2 могут формироваться пользователем с помощью оператора SET CHARACTER S, M. Для переключения с одного шрифта на другой используется оператор SET FONT n.

Среди других графических операторов, реализованных в составе Бейсик-Прос, отметим процедуру очистки экрана с одновременным сбросом текстового и графического курсора (CLEAR), оператор параллельного переноса графического изображения (SCROLL (ΔX, ΔY)) и серию процедур управления цветовой гаммой. Некоторые из последних операторов могут использоваться и на монохромном дисплее. Среди них разнообразные заливки замкнутой области (FILL), горизонтальной (FILLX) или вертикальной (FILLY) полос от заданной линии до границы ближайшей фигуры, средства управления способом отображения областей, имеющих общие части (SET WRITTING MODE) и др.

1. Ю ж а к о в А. Математическое обеспечение ЭВМ «Искра 226». — В кн.: Пакеты прикладных программ для микроЭВМ «Искра 266». — М.: МТЦНТИ. 1985. Вып. 28. С. 31—98.
2. Пу л Л. Работа на персональном компьютере. — М.: Мир, 1986.
3. Draft Proposed Standard for BASIC. ANSI X3J2/82—17. ANSI, 1982.
4. Draft International Standard ISO/DIS 7942: Graphics Kernel System. ISO, 1983.

Статья поступила 23 декабря 1986 г.

УДК 681.3

А. Е. Байков, А. Н. Выставкин, Б. М. Катков,
А. Э. Мартынов

ДИАЛОГОВЫЕ СРЕДСТВА ГРАФИЧЕСКОГО ВЗАИМОДЕЙСТВИЯ ДЛЯ СИСТЕМ АВТОМАТИЗАЦИИ ЭКСПЕРИМЕНТА

При разработке программного обеспечения систем автоматизации эксперимента (САЭ) на базе мини- и микроЭВМ значительная часть общих трудозатрат приходится на организацию диалогового взаимодействия экспериментатора с ЭВМ, в том числе графического. В ИРЭ АН СССР создана система, в которую входят инструментальные средства для разработки программного обеспечения взаимодействия в прикладных подсистемах (ППС) и набор диалоговых подсистем, решающих типовые задачи взаимодействия в САЭ (построение графиков, автоматизация чертежных работ, обработка экспериментальной информации и т. д.). Программные средства предназначены для работы на ЭВМ типа «Электроника 60», СМ-4 под управлением операционных систем РАФОС и ОС РВ и реализованы на языке PASCAL. Подсистемы обработки и отображения экспериментальной информации совместимы с системой ДИСПОГ [1, 2].

Ниже кратко перечислены основные части системы и их функциональные возможности.

1. Подсистема взаимодействия пользователя с ЭВМ — предназначена для организации взаимодействия во всех ППС, представляет собой инструментальный комплекс программных средств для:

- распознавания команд пользователя по минимальному числу литер;
- выявления неправильных и двусмысленных команд;
- выполнения командных файлов;
- ввода аргументов и проверки их корректности;
- вывода наводящих вопросов;
- выдачи справочной информации;
- ведения протокола взаимодействия с возможностью его последующего выполнения как командного файла;
- изменения режимов взаимодействия по командам пользователя;
- автоматическая замена схемы взаимодействия при загрузке новой ППС.

2. Базовое программное обеспечение машинной графики, ориентированное на применение САЭ и выполненное на основе принципов стандарта GKS:

- функции примитивов вывода;
- справочные функции;
- цвет;
- независимость от конкретного графического устройства;

- функции работы с метафайлами;
 функции преобразования координат.
3. Подсистема автоматизации чертежно-графических работ для САЭ:
 доступ в диалоговой форме к базовому ПО машинной графики;
 язык описания графической информации;
 реализация символов по ГОСТ 19.003-80 «Символы в схемах алгоритмов и программ»;
 хранение изображений в виде командного файла;
 возможность вывода графиков в поле чертежа.
4. Пакет прикладных программ (ППП) высокого уровня для построения зависимостей:
 пять типов осей координат;
 пять способов построения графиков (линия, точки, маркеры, линия + маркер, гистограмма);
 вывод текстовой информации в поле графика;
 вычерчивание разными типами линий;
 режим «окна» — изменение масштаба, выделение интересующих участков;
 три способа построения: из файла данных, из прикладной программы и «вручную»;
 возможность вывода трехмерных графиков при помощи составных файлов данных;
 вывод графиков в реальном времени эксперимента.
5. Диалоговая подсистема построения зависимостей, по своим возможностям аналогичная описанным в [1, 3]:
 реализация возможностей (4) в режиме диалога;
 возможность создания специальных командных файлов, задающих параметры графиков;
 оперативное отображение состояния системы (размеры окна, поля вывода, тип осей, графика, линии, маркера и т. д.).
6. Сервисная подсистема для работы с файлами данных:
 копирование, удаление, хранение, поиск;
 просмотр;
 преобразование в текст и обратно;
 автоматическая генерация тестовых файлов;
 автоматическое создание командного файла задания параметров графика.
7. Подсистема метафайлов:
 хранение готовых изображений;
 построение из метафайла;
 перенос изображения между системами;
 построение в режиме off-line (позволяет совмещать процесс вывода с работой других программ).
8. Подсистема базовой обработки:
 линейная интерполяция;
 приведение графика к равномерной сетке;
 приведение графиков к единой сетке;
 аппроксимация полиномами;
 сглаживание сплайнами;
 линейная фильтрация;
 удаление выбросов;
 операции над файлами данных (сложение, вычитание, базовые математические функции и т. д.).
10. Подсистема автоматизации разработки:
 разработка драйверов графических устройств;
 генерация схемы диалога прикладных подсистем;
 генерация рабочей конфигурации системы.

Телефон для справок: 460-24-42, Москва

ЛИТЕРАТУРА

1. Выставкин А. Н., Катков Б. М., Мартынов А. Э., Обухов Ю. В. Диалоговая система построения графиков в системах автоматизации экспериментов // Приборы и системы управления.—1986.— № 1.— С. 8—10.
2. Выставкин А. Н., Катков Б. М., Мартынов А. Э., Обухов Ю. В. Диалоговая система построения графиков в индивидуальных системах автоматизации экспериментов, Тезисы докладов Всесоюз-

ной конференции «ДИАЛОГ-84-МИКРО», Ленинград, 19—22 ноября, 1984.

3. Обухов Ю. В., Платонов С. А. Диалоговая система построения графиков в составе ИВСКП // Управляющие системы и машины.—1982.— С. 79—81.

Сообщение поступило 2 декабря 1986 г.

УДК 681.3.06

Ю. М. Баяковский, Т. Н. Михайлова, И. Г. Рыжова

РЕАЛИЗАЦИЯ БАЗОВОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ ДЛЯ ПЭВМ

Стандарт ГКС [1—3], утвержденный Международной организацией по стандартизации (ISO), определяет интерфейс между графической прикладной программой и набором функций, которые выполняет виртуальная графическая рабочая станция. Состав набора функций зависит от уровня ГКС. Стандарт предусматривает три уровня вывода (0, 1, 2) и три уровня ввода (а, б, с). Американский национальный стандарт [2] дополнен еще одним уровнем, который определен как результат исследования минимального графического интерфейса для программиста (PMIG).

Минимальный уровень ГКС (уровень μ) характеризуется наличием одной графической станции, минимальным набором подпрограмм вывода (четыре примитива: ломаная, полимаркер, полигональная область и текст), индивидуальным заданием атрибутов. В данном случае достаточно 48 функций (из них 18 справочных). Для сравнения можно отметить, что уровень 0а включает 92 функции (40 справочных).

Одна из первых реализаций минимального уровня [4] выполнена на языке СИ. Рассматриваемая в настоящей статье базовая графическая система мини-ГКС (соответствующая уровню μ) написана на языке Фортран. Все функции системы — это фортранские подпрограммы. Имена подпрограмм совпадают с предложенными в стандарте.

Для ЭВМ ЕС-1045 в системе виртуальных машин разработана экспериментальная версия мини-ГКС [5]. В версии для вывода графической информации на векторный дисплей ЕС-7064 и графопостроитель ЕС-7052 используются некоторые средства пакета БГП [6]. Затем был осуществлен перенос мини-ГКС на ПЭВМ типа ЕС-1841 с цветным графическим растровым дисплеем и матрично-точечным печатающим устройством.

При переносе возникли трудности, связанные с ограниченными возможностями языка и транслятора (отсутствуют возможности присвоения начальных значений элементам общих блоков, не допускаются операции с цепочками, инструкции ENTRY и т. д.). В связи с этим были внесены некоторые изменения в тексты фортранских программ. Кроме того, потребовалось написать драйверы для графических устройств.

Структура мини-ГКС

В структуре системы мини-ГКС (рис. 1) предусмотрена возможность распределенной ее реализации [7]. Мини-ГКС состоит из двух слов: инвариантного по отношению к устройствам и аппаратно-зависимого. К независимому от графической станции слою относятся доступные прикладной системе программы, реализующие функции ГКС, а также интерфейсная программа (диспетчер). В аппаратно-зависимый слой включены серверы и драйверы устройств.

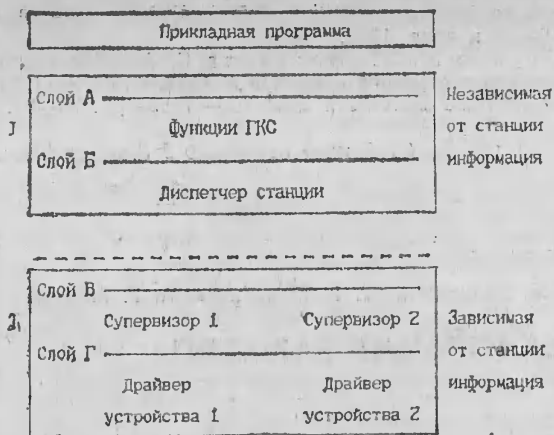


Рис. 1. Структура мини-ГКС

В слое, не зависящем от устройства, выполняется преобразование мировых координат в нормированные, формируется таблица состояния ГКС, осуществляется проверка состояния ГКС и выдаются диагностические сообщения.

Для связи не зависящего от устройства слоя и супервизора определен интерфейс, через который супервизору посылаются станционно-зависимые функции. Так как в мини-ГКС работает одна графическая станция, то интерфейсная программа достаточно простая. В зависимости от типа станции она вызывает необходимый в данный момент супервизор, в который передаются номер функции и ее параметры. Для передачи параметров имеются четыре массива фиксированного размера: целых чисел, два массива вещественных чисел и массив литер.

Супервизор выполняет 24 станционно-зависимые функции (открыть, закрыть, очистить, модифицировать станцию; функция расширения; примитивы: ломаная, полимаркер, текст, полигональная область; установить: тип ломаной, индекс цвета ломаной, тип полимаркера, индекс цвета полимаркера, индекс цвета текста, высоту литеры, выравнивание текста, вид заполнения полигональной области, индекс цвета полигональной области, представление цвета, прямоугольник отсеечения, индикатор отсеечения, окно станции, поле вывода станции; выяснить протяженность текста) и формирует таблицу состояния станции. Наличие или отсутствие отдельных функций определяется типом устройства (например, операции с цветами). С каждой функцией связан свой исполнитель. Для однородных устройств часть исполнителей супервизора фактически совпадает. Некоторые подпрограммы, например определение поля вывода станции с учетом подобия, отсеечение, преобразование нормированных координат (НК) в координаты устройства (КУ), формирование таблицы состояния станции, одинаковы во всех супервизорах.

Графическая информация, подготовленная супервизором, передается драйверу устройства, состоящему из нескольких ассемблерных программ (проведение отрезка, рисование маркера, написание литеры и др.).

Особенности реализации мини-ГКС

Супервизор и драйвер реализуются с учетом специфических особенностей графических устройств. Цветной графический растровый дисплей в режиме среднего разрешения имеет следующие характеристики: растр 320×200 пикселей, 16 цветов фона, 2 палитры по 4 цвета в каждой для засветки пикселей (включая цвет фона), аппаратный генератор символов с кодировкой для одной литеры в растре 8×8.

Используемое матрично-точечное печатающее устройство (EPSON FX-85) имеет по ширине каретки (по горизонтали) 480 точек. Расстояние между точками по горизонтали 1/60 дюйма, по вертикали — 1/72 дюйма. Цвета отсутствуют.

Характеристики устройств содержатся в таблице описания графической станции. Таблица разделена на две части: в одной из них хранятся вещественные величины, в другой — целые. Таблицы станций занимают два массива — WDT и IWDT:

Массив IWDT (18,3) для каждой станции содержит:

IWTVP	— тип рабочей станции
IDC	— единицы системы координат устройства
INRAST (2)	— максимальные размеры поверхности отображения в растровых единицах
IDEV	— растровый или векторный дисплей
ICOLRP	— представление цвета (невявно/немедленно)
KTR	— преобразование станции
IRM	— значение по умолчанию для невявной регенерации
LTVP	— число возможных типов линии
LINTVP (4)	— список возможных типов линии
NCHHT	— число возможных высот литеры
NCL	— число возможных цветов
ICOL	— доступность цвета
NPRCI	— число предварительно определенных индексов цвета
NMXCI	— максимальное число индексов цвета.

Массив WDT (8,3) содержит:

DCUN (2)	— максимальные размеры поверхности отображения в единицах координат устройства
TABCL (4, 3)	— таблица предварительно определенных индексов цвета.

Функции RDWDT и IRDWDT предназначены для чтения информации из этих массивов:

```

FUNCTION IRDWDT(NTAB, NCRUP, NEL)
  INTEGER NTAB, NCRUP, NEL, IPOINT(14), IWDT(18, 3)
  DATA IPOINT /0, 1, 2, 4, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17/
  DATA IWDT /1, 0, 320, 200, 1, 1, 2*0, 4, 5*1, 4, 0, 4, 4,
*           2, 0, 479, 720, 1, 1, 2*0, 4, 5*1, 2, 1, 2, 2,
*           3, 17*0/
  I=IPOINT(NCRUP)
  IRDWDT=IWDT(I+NEL, NTAB)
  RETURN
END

FUNCTION RDWDT(NTAB, NCRUP, NEL)
  INTEGER NTAB, NCRUP, NEL, IPOINT(2)
  REAL WDT(14, 3)
  DATA WDT /0.320, 0.200, 3*0., 3*1., 3*1., 3*1.,
*           0.200, 0.250, 5*0., 1., 5*0., 1.,
*           0.200, 0.200, 12*0./
  DATA IPOINT /0, 2/
  I=IPOINT(NCRUP)
  RDWDT=WDT(I+NEL, NTAB)
  RETURN
END

```

ОДИН ИЗ ПЕРВЫХ ЦЕНТРОВ ИНФОРМАЦИОННЫХ УСЛУГ ОТКРЫЛСЯ В МОСКВЕ



Центр: сдает в аренду ПЭВМ, проводит обучение групп и отдельных граждан, продает пакеты программ для ПЭВМ и покупает их у населения.
Телефон для справок: 242-88-65.

МАШИННАЯ ГРАФИКА

Архитектура современных вычислительных систем и уровень развития элементной базы позволяют создавать сравнительно недорогие малогабаритные устройства обработки полутоновых изображений, значительно расширяющие возможности серийных ПЭВМ. Малая дисплейная система (МДС) «Кама-М», совместимая по протоколу обмена сигналами с си-

стемной магистралью И41, предоставляет в распоряжение пользователя память изображений объемом 1М байт (4 кадра размерами $512 \times 512 \times 1$ байт) и интерфейсы сопряжения со средствами ввода/вывода изображений: телекамерой, RGB-монитором, фототелеграфной аппаратурой и т. п. Интерактивное управление режимами работы МДС осуществляет-



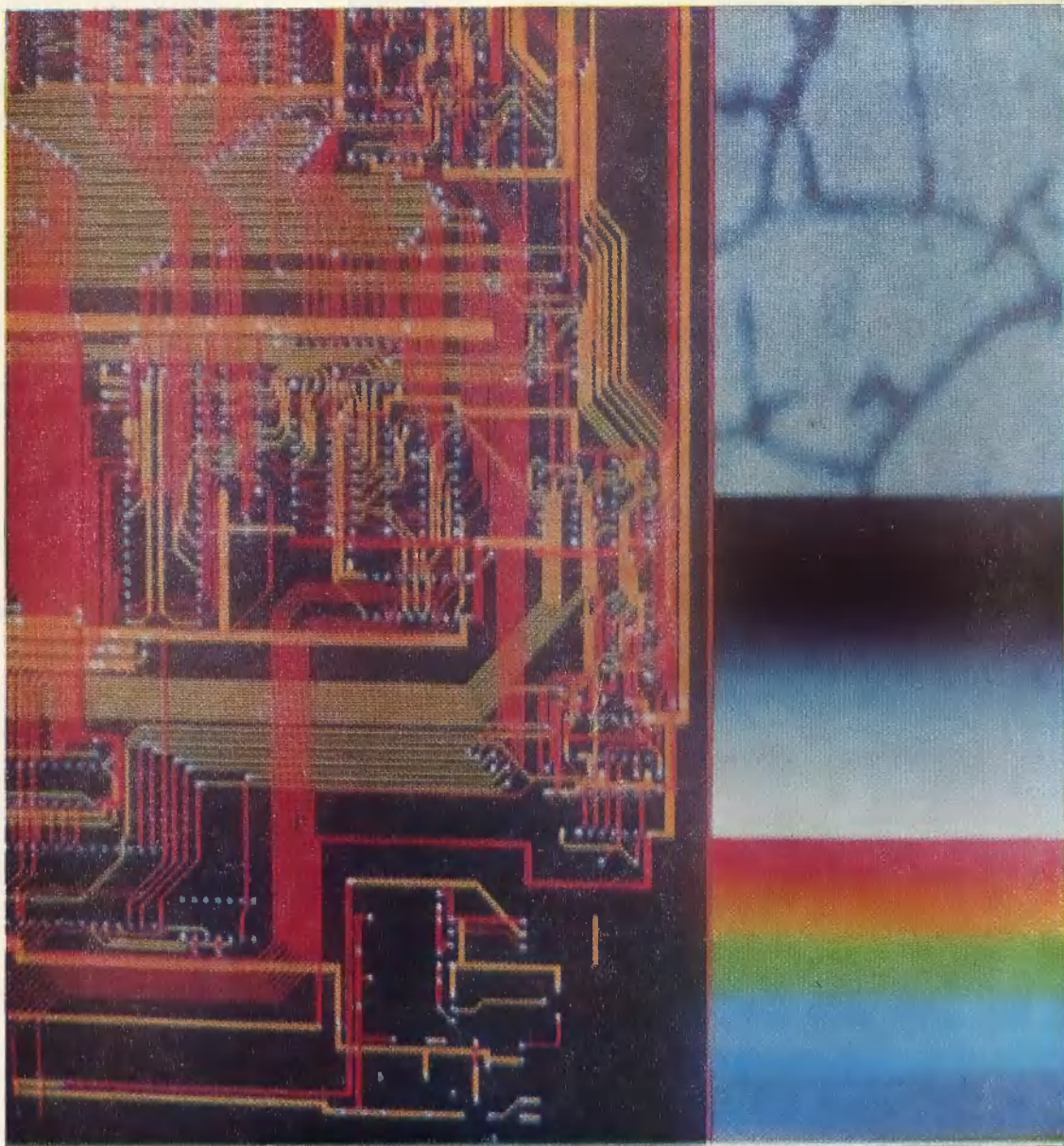
Комбинированное изображение, полученное путем ввода через телекамеру нескольких фрагментов с последующей перекраской в условные цвета

МАШИННАЯ ГРАФИКА

ся путем программирования и опроса соответствующих регистров.

Программное обеспечение системы ориентировано на решение задач машинной графики и обработки данных дистанционного зондирования. Это обуславливает возможность применения МДС в природопользовании, медицине, биологии, автоматизированном

проектировании и т. п. Кроме того, МДС позволяет производить синтез изображений в условных цветах, трансформацию, классификацию, сегментацию и измерение характеристик изображаемых объектов. Для формирования и коррекции топологии многослойных печатных плат разработан специализированный цветовой графический редактор.



Фрагмент печатной платы и клеточная структура. Гамму цветов можно подобрать в интерактивном режиме по отображаемому цветовому клину

ТЕРМИНАЛ УДАЛЕННОГО ДОСТУПА

Удаленный процесс-терминал РПТ-80 разработан для:

измерения большого числа физических величин;

осуществления операций, связанных с обработкой данных измерений;

поддержания связи с другими элементами сети управления.

Процесс-терминал имеет модульное построение, содержит микропрограммируемый блок управления. На шину блока управления подключаются блоки управления памяти, блоки сопряжения внешних устройств и каналов передачи данных, а также блоки, служащие для подключения сигналов технологического процесса.



Структура процесс-терминала обеспечивает возможность параллельной работы нескольких вспомогательных процессоров наряду с центральным процессором (например, вспомогательный процессор передачи данных), повышает эффективность работы терминала.

Программное обеспечение:

операционная система RPS для программирования параллельных процессов (мультизадачность) это: элементарные вспомогательные средства;

пакет арифметических подпрограмм;

подсистема ввода/вывода; программы передачи данных;

пакет пользовательских программ;

система разработки программ UPM и т. д.

Услуги, предоставляемые фирмой:

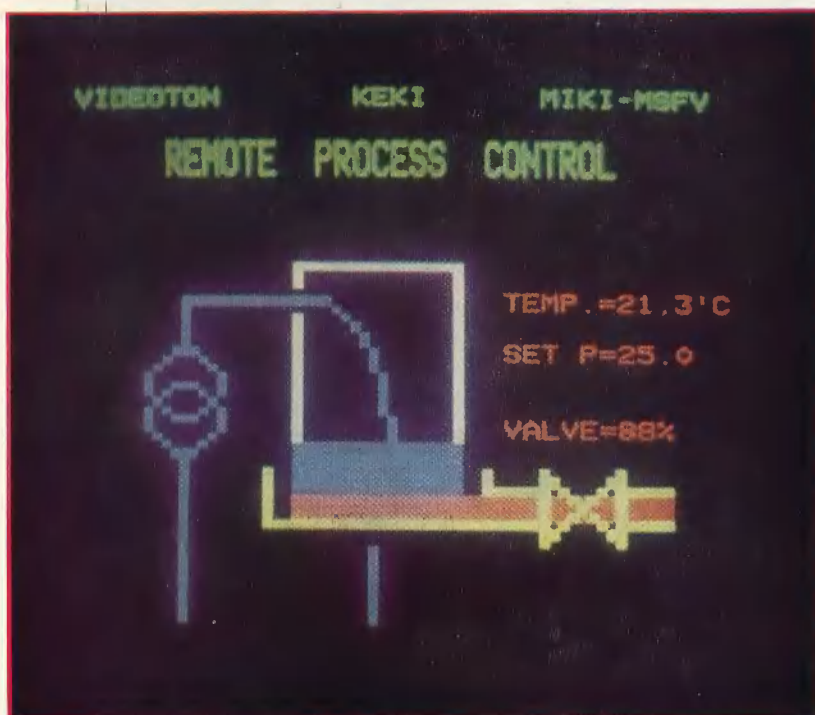
анализ задач, проектирование систем управления процессами;

создание комплектных промышленных систем;

проектирование и создание диспетчерских центров и т. д.

Основные области применения:

сбор данных измерения; управление процессом; автоматизация.



Будапешт. Завод вычислительной техники, 11.

Каждый из исполнителей супервизора завершает выполнение определенной функции ГКС, зависящей от станции. Так, при открытии графической станции формируется таблица состояния станции, запоминается переданная информация об атрибутах во внутренних общих блоках, инициализируется работа драйвера.

Для ломаной, например, в супервизор передается информация о ее вершинах в нормированных координатах. Обработывается ломаная по отрезкам. С помощью программы отсечения определяется положение отрезка относительно отсекающего прямоугольника. Координаты отрезка или его части, попадающей внутрь отсекающего прямоугольника, преобразуются в координаты устройства. Затем КУ преобразуется в целочисленные координаты устройства, и отрезок в этих координатах передается драйверу устройства.

По аналогичному пути обрабатывается информация для полимаркера. В программе заполнения полигональной области используется подпрограмма рисования ломаной.

Каждый из примитивов имеет набор индивидуально устанавливаемых атрибутов. Их значения хранятся внутри таблицы состояния ГКС, передаются супервизору при открытии соответствующей станции и изменяются при обращении к программам установки атрибутов. Программно реализованы следующие атрибуты примитивов для:

ломаной — тип линии: сплошная, штриховая, пунктирная, штрихпунктирная, штрихпунктирпунктирная (рис. 2, а);

полигональной области — вид заполнения: пусто, прямоугольная сетка, двойная штриховка с линиями наклона 45 и -45° , штриховка с линиями наклона 45° (рис. 2, б);

полимаркера — тип маркера: точка, плюс, звездочка, кружок и крестик (рис. 2, в);

текста — горизонтальное выравнивание: левое, центральное, правое (рис. 2, г).

Каждый из примитивов может иметь атрибут — индекс цвета. В зависимости от индекса цвета по итен-

сивностям в таблице цветности определяется конкретный цвет примитива.

Драйвер. Каждый из драйверов устройств состоит из нескольких подпрограмм, написанных на языке ассемблера. Для матрично-точечного печатающего устройства (принтера) драйвер включает в себя программы инициализации устройства, проведения отрезка, рисования маркера и литер, терминирования устройства. Аналогичные функции выполняет и драйвер дисплея. Кроме того, имеются дополнительные программы, связанные цветом. Занесение информации в соответствующие регистры и в память дисплея или принтера осуществляется с использованием прерываний INT 10 и INT 17, реализованных в базовой системе ввода-вывода.

В программах проведения отрезка применяется алгоритм Брезенхема [8]. По вычисленным координатам пикселей заполняется битовая карта в случае дисплея или формируется матрица изображения в оперативной памяти для последующего вывода на принтер. В программе для рисования маркеров используется кодировка на сетке 5×5 , для рисования литер — кодировка на сетке 8×8 .

Как уже говорилось выше, имеются две реализации мини-ГКС для ЕС-1045 и ЕС-1841. Объем каждой реализации примерно 3500 строк на Фортране, из них 520 строк — комментарии (верхний уровень — 2300, нижний — 1200 строк). Ассемблерная часть для ЕС-1841 составляет около 1000 предложений. При работе на ЕС-1841 требуется память примерно 110...120К байт.

На рис. 3, а показаны некоторые возможности формирования рисунка средствами мини-ГКС. Соответствующая программа приведена на рис. 3, б.

Графор-ГКС

ГКС — базовая графическая система, не содержащая графических функций высокого уровня, таких как изолинии, поверхности и т. д. В связи с этим возникает задача создания новых прикладных графических систем, базой для которых является ГКС, или привязки к ГКС имеющихся систем.

Авторами была сделана попытка использовать ГКС в качестве базового уровня системы Графор [9]. Работа выполнялась на ЕС-1045 в системе виртуальных машин. При этом рассматривались различные варианты подключения Графора к ГКС. Некоторые из них потребовали расширения мини-ГКС, в частности, увеличения числа преобразований до трех (с учетом инвариантного нулевого преобразования).

В программах Графора для проведения линии используется программа MOVE, рисующая отрезок из

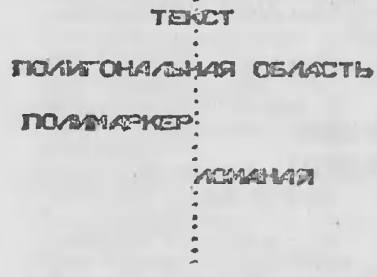
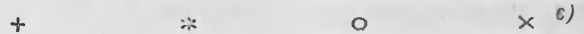
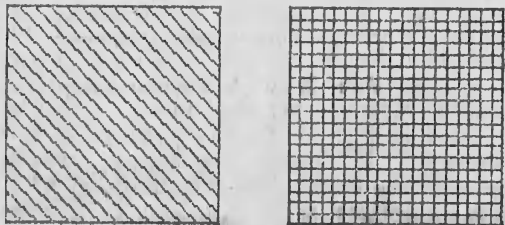
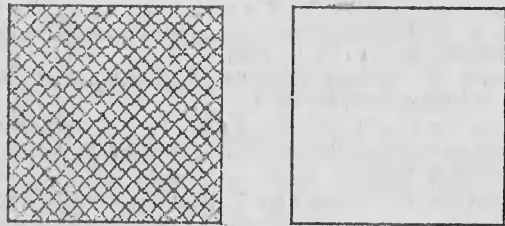
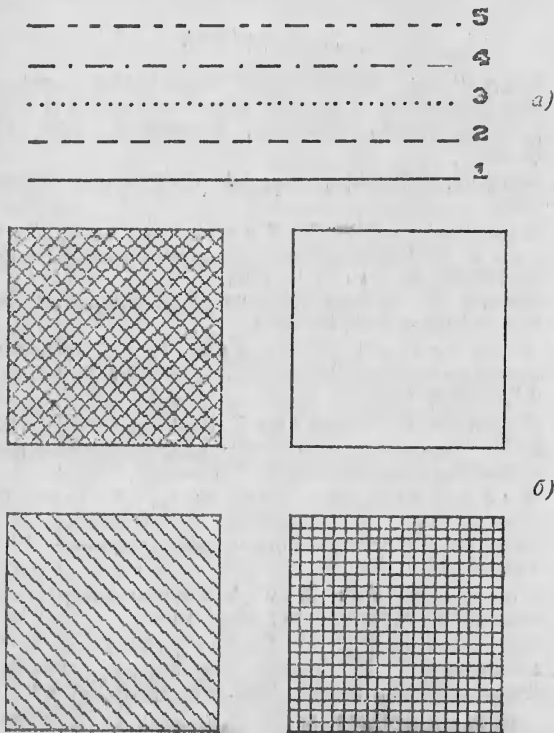
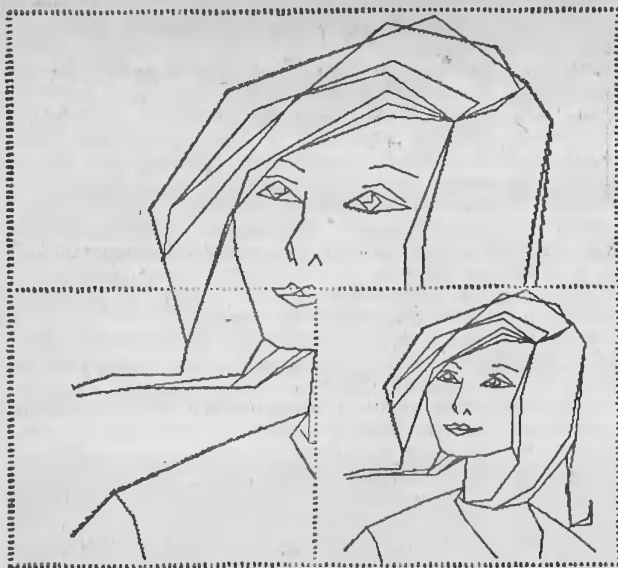


Рис. 2. Атрибуты ломаной (а), полигональной области (б), полимаркера (в), текста (г)



a)

```

REAL X1(5), Y1(5), X2(5), Y2(5), X3(5), Y3(5)
DATA X1/0., 10., 10., 0., 0./, Y1/0., 0., 9., 9., 0./
DATA X2/0., 5., 5., 0., 0./, Y2/0., 0., 4.5, 4.5, 0./
DATA X3/0., 10., 10., 0., 0./, Y3/4.5, 4.5, 9., 9., 4.5/
DATA IWKID, ICONID, IRRFIL/15, 10, 5/
WRITE (*, *) ' IUTYPE='
READ(*, *) IUTYPE
CALL GOPKS( IRRFIL )
CALL GOPWK( IWKID, ICONID, IUTYPE )
CALL GACWK( IWKID )
CALL GSWN(1, 0., 10., 0., 9.)
CALL GSWP(1, 0.5, 1., 0., 0.45)
CALL GSLN(3)
CALL GPL(5, X1, Y1)
CALL GIRL
CALL GSWN(1, 0., 5., 0., 4.5)
CALL GSWP(1, 0., 0.5, 0., 0.45)
CALL GSLN(3)
CALL GPL(5, X2, Y2)
CALL GIRL
CALL GSWN(1, 0., 10., 4.5, 9.)
CALL GSWP(1, 0., 1., 0.45, 0.9)
CALL GSLN(3)
CALL GPL(5, X3, Y3)
CALL GIRL
CALL GDAWK( IWKID )
CALL GCLWK( IWKID )
CALL GCLKS
STOP
END

```

b)

текущей точки в заданную или меняющая значение текущей позиции (для графопостроителя это переход из текущей позиции в заданную с опущенным или поднятым пером). При переработке программ Графора замена обращений к MOVE, LINEO и др. на обращения к программе GPL приводит к потере некоторых возможностей, имеющихся в Графоре: аффинные преобразования, экранирование, след пера. Причина в том, что в Графоре преобразования, задаваемые соответствующими программами, осуществляются на нижнем уровне в программах MOVE1, MOVE2, MOVE3.

Кроме того, нельзя непосредственно использовать программы, работающие с полярными координатами, так как полярная область не является прямоугольной.

Аффинные преобразования удалось ввести с помощью нестандартных функций (ESCAPE). Эффективная реализация некоторых других возможностей Графора (в частности, экранирования и следа пера) требует серьезной переработки и даже применения иных алгоритмов.

Во многих программах Графора существенно используется понятие текущей позиции (пера, луча). Она, в частности, может определять один из концов отрезка, центр окружности и т. п. В мини-ГКС понятие текущей позиции отсутствует. Поэтому при объединении Графора и ГКС в отдельных случаях потребуется вносить изменения в прикладную программу.

В простейшем варианте привязки Графора к ГКС можно обойтись одним преобразованием, обращаясь к программе рисования ломаной с координатами отрезка. Все возможности Графора при этом сохраняются, но время работы программ Графора настолько увеличивается, что использование комбинации Графор-ГКС становится неэффективным.

Наш опыт показывает, что подключение к ГКС графической системы, имеющей свой развитый базовый уровень, вряд ли оправдано. По-видимому, целесообразна разработка новых систем, в которых в полной мере учитываются и используются средства, имеющиеся в ГКС.

Телефон для справок: 333-71-89, Москва

ЛИТЕРАТУРА

1. International standard ISO 7942—1985(E). Information processing. Computer graphics. Graphical Kernel Systems (GKS) functional description.— ISO, 1985—08-15.
2. Computer Graphics. Special GKS issue. February, 1984.
3. Баяковский Ю. М., Галактионов В. А., Кудин Б. В. Графические стандарты. Обзор. Препринт № 155 ИПМ АН СССР, 1984.
4. Randall W. Simons. Minimal GKS//Computer Graphics.— 1983.— Vol. 17, № 3.
5. Михайлова Т. Н., Рыжова И. Г. Базовая графическая система. Мини-ГКС. Препринт № 163 ИПМ АН СССР, 1986.
6. Ланис А. О., Лебедев Г. В., Романенко С. А. БГП — базовый графический пакет. Функциональное описание. Препринт ИПМ АН СССР, 1983.
7. Waggoner C. N., Tucker C., Nelson C. J. NOVA * GKS, A Distributed Implementation of the Graphical Kernel System//Computer Graphics.— 1984.— Vol. 18, № 3.
8. Фоли Дж., ван Дэм А. Основы интерактивной машинной графики.— М.: Мир, 1985.
9. Баяковский Ю. М., Галактионов В. А., Михайлова Т. Н. Графор. Графическое расширение фортрана.— М.: Наука, 1985.

Статья поступила 7 августа 1987 г.

Рис. 3. Пример формирования рисунка средствами ГКС

СТАНДАРТИЗАЦИЯ ИНТЕРФЕЙСОВ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

ОБОЗНАЧЕНИЕ И НАЗНАЧЕНИЕ СИГНАЛОВ В ФИРМЕННЫХ
И ГОСУДАРСТВЕННЫХ СТАНДАРТАХ НА ИНТЕРФЕЙС ДЛЯ МММС

A	Address	ША Адрес
AC	Arbitration Condition	Условие при арбитраже
ACCLK	Alternating Current Clock	Синхронизация от источника переменного тока
ACC REQ	Access Request	Запрос на доступ к магистрали
AC FAIL	Alternating Current Fail	Неисправность источника переменного тока
ACI	Auxillary Priority Chain In	Вход дополнительной цепочки приоритетов
ACK	Acknowledge	ОТВ, ПЗ, ППД. Подтверждение передачи данных
ACK IN	Acknowledge In	Подтверждение по входу
ACK OUT	Acknowledge Out	Подтверждение по выходу
ACLO	Alternating Current Low	Напряжение переменного тока ниже нормы
ACO	Auxillary Priority Chain Out	Выход дополнительной цепочки приоритетов
AD	Address / Data	Адрес/данные
ADI	Address/Data/Interrupt	Адрес/данные/прерывания
ADR STB	Address Strobe	Строб адреса
ADSB	Address Disable Signal Bus	Сигнал несоответствия адреса по шине
AEN	Address Enable	Разрешение адресации
AERR	Address Error	Ошибка в адресе
AG	Arbitration Grant	РМД. Разрешение арбитража
AI	Address Acknowledge Inverse	Инверсный сигнал подтверждения адреса
AI	Arbitration Request Inhibit	Запрет запроса на арбитраж
AK	Address Acknowledge	ППА. Подтверждение приема адреса
AL	Arbitration Level	ЗМ. Уровни арбитража
ALATCH	Address Latch	Запрет обращения к адресам
ALE	Address Latch Enable	Сигнал записи адреса
AMX	Address Modifier	Линии модификации адреса X <7...0> для VERSABUS и X <5...0> для VME-BUS
AN	Arbitration Number	ЭДМ. Число линий арбитража
AP	Arbitration Priority	Приоритет арбитража
APARITY	Address Parity	Четность адреса
AQ	Acquired	Захват магистрали
AR	Arbitration Request	Запрос при арбитраже
ARB	Arbitration Bus	Линии арбитража
ARBCLK	Arbitration CLK	Синхронизация арбитража
AREADY	Advance Ready	Ускоренная готовность
AS	Address Strobe	Строб адреса
AUX RESET	Auxiliary Reset	Сброс при включении и восстановлении электропитания
AVE	Address Valid Executive	Достоверность установления адреса
AUX	Auxiliary	Резервный источник питания
BA	BUS Available	Разрешение магистрали
BACK IN, BAI	BUS Acknowledge IN	Подтверждение прерывания по входу
BACK OUT, BAO	BUS Acknowledge Out	Подтверждение прерывания по выходу
BAT	Battery	Аккумуляторная батарея
BB, BBSY, BUSY	BUS Busy	ЗАН. Магистраль занята
BS	I/O Select Signal	Выбор устройства в/в
B/W	Byte / Word	Работа байтами/словами
BCLK, BUSCL	BUS Clock	СИНХР, ТЧШ. Линия синхронизации магистрали
BCLR	BUS Clear	Выборочный сброс памяти
BDIN	BUS Data In	Управление вводом данных
BDRDY	BUS Data Ready	Готовность данных
BERR, BUS ERR	BUS Error	Ошибка на магистрали
BG, BGRT, BUS Gr	BUS Grant	РЗМ. Разрешение магистрали
BGACK BUS	Grant acknowledge	Подтверждение разрешения на пользование магистралью
BGIN, BGTI	BUS Grant In	РЗМвх, ПРПП. Разрешение магистрали по входу

BGOUT, BGTO	BUS Grant Out	РЗМвых, ПРРИ. Разрешение магистрали по выходу
BH	BUS Halted	Магистраль остановлена
BHE, BHEN	Byte High Enable	ПЗМ, ПЗМП, ПЗМИ. Разрешение работы байтами
BP	BUS Priority	Линия приоритетов
BPAR	BUS Parity	Линия четности
BPRN	BUS Priority In	Вход линии приоритетов
BPRO	BUS Priority Out	Выход линии приоритетов
BPRQ	BUS Priority Request	Запрос линии приоритетов
BPS	BUS Priority Scheduler	Уровни приоритетов
BR, BREQ, BUSRQ,	BUS Request	ЗМ, ЗПШ. Запрос магистрали
BUSRQ		
BREL	BUS Release	Освобождение магистрали
BTMO	BUS Time out	Перерыв на магистрали при неисправности
BUSACQ	BUS Acquired	Захват магистрали
BUS AK	BUS Acknowledge	ПМА. Линия подтверждения приема адреса
BUS Deal	BUS De-allocate	Управление переключением магистрали
BUS EN	BUS Enable	Линия разрешения работы
Byt Ad	Byte Address	ПЗП. Работа байтами адресов
Byt Wk	Byte Working	Работа байтами данных
CX	Command	Линия управления X <1...0> для ОШ и X <2...0> для К-896
CAI	CPU Acknowledge In	Вход подтверждения процессора
CAO	CPU Acknowledge Out	Выход подтверждения процессора
CARD SLCTD	Card Selected	Выбор модуля
CAVAIL	CPU Available	Разрешение процессору доступ к магистрали
CBRQ	Common Bus Request	Общий запрос к магистрали
CcAb	Cycle Abort	Останов цикла
CcBn	Cycle Begin	Начало цикла
CcFin	Cycle Finish	Окончание цикла
CCI.K	Constant Clock	ОФЧ. Постоянная частота синхронизации
Cc Res	Cycle Response	Ответ
CDSB	Control Disable	Сигнал несоответствия кода управления на магистрали
CHI	Signal Bus	Вход цепи приоритетов
CHO	Chain In	Выход цепи приоритетов
CK, CLK	Chain Out	Синхронизация
CMX	Clock	Линия управления X <4...0>
CP	Command	Управление четностью
CPUREQ	Command Parity	Запрос процессора
CS	CPU Request	Выбор прибора
D	Cristal select	ПД. Данные
DACK, DXACK	Data	Подтверждение ПДП
DAT STB	DMA Acknowledge	Строб данных
DBIN	Data Strobe	Управление вводом данных
DC FALL	Data Bus In	Неисправность электропитания постоянного тока
DCLOW	Direct Current	Напряжение источника постоянного тока ниже нормы
DCOK	Power Fail	АИП. Источник постоянного тока исправен
DDS	Direct Current	
DEBUG	Power Low	
DEN	Direct Current	
DERR	Power Okay	
DEVSEL,	Data Direct Signal	Синхронизация данных при ПДП
DESELECT	Debug	Поиск неисправности
DI	Data Enable	Разрешение данных
DIN	Data Error	Ошибка данных
DK	Dewice Select	Выбор устройства
DMA	Data Acknowledge	
DMACK	Invers	Инверсный сигнал подтверждения данных
DMA IN	Data In	
DMAOUT	Data Acknowledge	ДЗП. Управление вводом данных
DMARDY	Direct Memory Acces	НД, ОТВ. Подтверждение приема данных
DMA RQ, DMR	DMA Acknowledge	Прямой доступ к памяти (ПДП)
DMGI	DMA In	Подтверждение ПДП
	DMA Out	Вход ПДП
	DMA Ready	Выход ПДП
	DMA Request	Готовность ПДП
	DMA Grant In	Запрос ПДП
		Разрешение ПДП по входу

DMGO	DMA Grant Out	Разрешение ПДП по выходу
DOD SB	Data Outrut Disable	Сигнал несоответствия кода в выходных данных
DOUT	Signal Bus	ДЧТ. Управление выводом данных
DPAR, DPARITY	Data Out	Четность данных
DRQ	Data Parity	Запрос ПДП
DRQ, DREQ	DMA Request	Запрос ПДП
DS	Data Strobe	Строб данных
DTACK	Data Transfer Acknowledge	ОТВ. Подтверждение передачи данных
DXACK	DMA Transfer Acknowledge	Подтверждение передачи данных при ПДП
EC	Error Control	Определение и коррекция ошибки
ED	Error Direction	Определение ошибок
EG	Enable Geographical	Разрешение географической адресации
EN	Enable	Разрешение
ENF	Enable Final	Разрешение окончания
ENI	Enable Interrupt	Разрешение прерывания
EV	Error Valid	ОШВ. Достоверность ошибки
EVNT	External Event	Линия внешнего события
EWR	Early Write	Предварительная запись
FIRQ	Fast Interrupt Request	Быстрый запрос прерывания
GA	Geographical Address	Географический адрес
G/H	Go/Halt	Выполнение/Останов
GK	Grant Acknowledge	Подтверждение разрешения на арбитраж
GRI	Grant In	Разрешение прерывания по входу
GRO	Grant Out	Разрешение прерывания по выходу
HALT, HLT	Halt	ОСТ. Останов при ПДП
HALT ACK	Halt Acknowledge	Подтверждение останова
HBS	High Byte Strobe	Строб записи старшего байта
HLDA	Hold Acknowledge	Подтверждение захвата магистрали при ПДП
HOLD	Hold	Захват магистрали при ПДП
HOLDA	Hold Acknowledge	Подтверждение захвата магистрали при ПДП
IACK IN	Interrupt Acknowledge In	Подтверждение прерывания по входу
IACK OT, IAO	Interrupt Acknowledge Out	Подтверждение прерывания по выходу
ID	Identification Device	Идентификатор устройства
IDLATCH	Identification Device Latch	Блокировка устройства идентификации
IEI	Interrupt Enable In	РПШвх. Разрешение прерывания по входу
IEO	Interrupt Enable Out	РПШвых. Разрешение прерывания по выходу
INH	Inhibit Line	ЗАПР. Линия запрета
INH1	Inhibit RAM Signal	Запрет обращения к ОЗУ
INH2	Inhibit ROM Signal	Запрет обращения к ПЗУ и ППЗУ
INIT	Initialization Signal	Сигнал инициализации
INT, IT	Interrupt	ЗПР, ПРЕР. Прерывание
INTA, INTAK, IVTA	Interrupt Acknowledge	Подтверждение прерывания
INTAE	Interrupt Acknowledge Enable	Разрешение подтверждения прерывания
INTEN	Interrupt Enable	Расширение прерывания
INTR, INTRQ, IRQ	Interrupt Request	ЗПР. Запрос прерывания
INTREXP	Interrupt Request Enable	Разрешение запроса на прерывание
INTREXP	Interrupt Request Expansion	Расширение запросов на прерывание
IO CHRDY	I/O Channel Ready	Готовность канала в/в
IO CLCK	I/O Channel Check	Проверка канала в/в
IOEXP	I/O Expansion	Расширение в/в
IOM	I/O Memory	ПВУ. Работа с в/в или памятью
I/OR	I/O Read	ПРМ, ЧТЭ. Чтение устройств в/в
IORC	I/O Read Command	Чтение устройств в/в
IORQ	I/O Request	ЗД. Запрос устройств в/в
IORST	I/O Reset	Сброс устройств в/в
IOSEL	I/O Select	ВУ. Выбор устройств в/в
I/O STR	I/O Strobe	Строб в/в
I/OW	I/O Write	ВДЧ, ЗПЭ. Запись в устройства в/в
IOWC	I/O Write Command	Запись в устройства в/в

IPRQ	Inter Processor Request	Межпроцессорный запрос
LACH	Lach	Блокировка идентификатора
LACK	Looped Interrupt Acknowledge	Начало цепи подтверждения прерывания
LAS	Lower Address Strobe	Строб записи младших разрядов адреса
LBS	Lower Byte Strobe	Строб записи младшего байта
LDS	Lower Data Strobe	Строб записи данных младших разрядов
LMS	Local master/slave	Место ведущего/ведомого
LOCK	Lock	Блокировка магистрали при двухпортовой работе
LWORD	Long Word	Управление длиной слова
LXACK	Local Acknowledge	Местное подтверждение передачи
MACK	Memory Acknowledge	Подтверждение доступа к памяти
MCLK	Master Clock	Синхронизация ведущего
MC SYNC	CPU Machine Cycle Sync	Машинный цикл синхронизации процессора
MEM	Memory	Чтение или запись в память
MEM CLK	Memory Clock	Синхронизация динамической памяти
MEMEN	Memory Enable	Разрешение доступа к памяти
MEMEX	Memory Expansion	Расширение памяти
MEM/IO, M/IO	Memory/IO	Работа с памятью или в/в
MEMR, MRDC	Memory Read	ЧТН, ЧТЗУ. Чтение памяти
MEMRDY	Memory Ready	Готовность памяти
MEMRQ, MRQ, MREQ	Memory Request	Запрос памяти
MEMW, MWTC	Memory Write	ЗАП, ЗПЗУ. Запись в память
MEMWID	Memory Width	Расширение памяти
MMAI	Multimicro Acknowledge In	Подтверждение по входу при многопроцессорной работе
MMAO	Multimicro Acknowledge Out	Подтверждение по выходу при многопроцессорной работе
MMRAP	Memory Mapping Code	Кодирование пространства памяти
MMREQ	Multimicro Request	Запрос магистрали при многопроцессорной работе
MPRO	Memory Protect	Защита памяти
MSX	Mode Select	Выбор режима X <2...0>
MSYN	Master Sync	Синхронизация ведущего
NMI	Nonmaskable Interrupt	ГВС, ПРС. Не маскируемое прерывание
NMIRQ	Nonmaskable Interrupt Request	Не маскируемый запрос на прерывание
NPG	Nonprocessor Grant	Разрешение не процессорного прерывания
NPR	Nonprocessor Request	Не процессорный запрос
N/S	Normal/System	Управление пользователем или системой
P, PAR	Parity	Четность
PA	Parity Available	Наличие разряда четности
PB	Parity Bit	Разряд четности
PBRESET	Push-button Reset	Сброс кнопочный
PCI	Priority Chain In	Вход цепи приоритетов
PCO	Priority Chain Out	Выход цепи приоритетов
PE, PARITY-ERR	Parity Error	ОШВ. Ошибка по четности
PE	Parity Enable	Разрешение контроля по четности
PF	Power Fail	Отказ электропитания
PFD	Power Fail Direct Current	Отказ источника постоянного тока
PFI, PFIN	Power Fail Interrupt	Прерывание по неисправности электропитания
PFL	Power Fail Low	Напряжение электропитания ниже нормы
PFSN	Power Fail Sense	Неисправно электропитание
PFSR	Power Fail Sense Reset	Сброс сигнала неисправности электропитания
POC	Power-on Clear Signal	Сброс при включении электропитания
POI	Power Interrupt	Прерывание по неисправности электропитания
POK	Power Okay	АСП. Исправно электропитание
PRESET	Push Reset	Сброс кнопочный
PRESET	Periferical Reset	Сброс периферийных устройств
PROT	Protect	Защита при неисправности электропитания
PWRBAD	Power Bad	Повреждение электропитания
PWR FAIL, PWE, PF,	Power Fail	Повреждение электропитания
PFAIL		
PWRS	Power Reset	Сброс сигнала неисправности электропитания
PWRCL	Power Clear	Сброс сигнала неисправности электропитания
R, RD	Read	Чтение

RB, RE, RS	Reset Bus	СБР, УСТ. Сброс магистрали
RBB	Receiver Busy Bus	Приемник линии занятости
RDY	Ready	Готовность
REF, RFSH, REFRESH	Refresh	Регенерация динамической памяти (РДП)
REFCLK	Refresh Clock	Синхронизация при РДП
REF GR	Refresh Grant	Разрешение на РДП
REFREQ	Refresh Request	Запрос на РДП
REPLY	Reply	Ответ устройства
RES TART	Reset Start	Повторный запуск
RO	Reset	Сброс
RQ, RQST	Request	Запрос на обслуживание
RSTNC	Reset Not Completed	Выборочный сброс
R/W	Read/Write	Чтение/запись
RX	Receive Serial Line	Приемная последовательная линия
SA	Schalt agresse	Адрес коммутации
SACK	Selection Acknowledge	Подтверждение выбора
SB	Serial Bus	Последовательная магистраль
SC	Supervisor	Супервизор управления
SCX	Signal Control	Сигналы управления X <9 ... 0>
SDCB	Status Disable Signal Bus	Сигнал несоответствия кода на линиях со- стояния
SECIEN	Secondary Interrupt	Разрешение вторичного прерывания
SECRESET	Enable	Выборочный сброс
SER CLK	Selective Reset	Синхронизация последовательной магистра- ли
SER DAT	Serial Clock	Передача данных по последовательной ма- гистрале
SERL	Serial Data	Последовательная связь
SIORREQ	Serial Link	Сигнал запроса в/в
SP	Signal I/O Request	Сигнал четности
SPV	Signal Parity	Достоверность ошибки по четности
SR, SRQ	Signal Valid	Запрос на обслуживание
SSX	Service Request	Состояние ведомого X <2 ... 0>
SSYN	Slave Status	Синхронизация ведомого
STDBY	Slave Sync	Резервный источник питания
STOP	Standby	Останов
STX	Stop	Линии состояния X <2 ... 0> для FUTURE- BUS и X <4 ... 0> для Z-BUS
SWRITE	Status Lines	Сигнал записи/чтения
SYCLK, SYSCLK	Signal Write/Read	Системная синхронизация
SYNC	System Clock	Линия синхронизации
SYSFALL	Sync	Системный тест
SYSRESET	System Fail	Системный сброс
TIMOUT	System Reset	Ожидание
TMRFS	Time Out	Надежный отметчик времени
TMX	Timer Fail Safe	Режим передачи X <1 ... 0>
TX	Transfer Mode	Передающая последовательная линия
UAS	Transmit Serial Line	Строб старших разрядов адреса
UDS	Upper Address Strobe	Строб данных старших разрядов
VED	Upper Data Strobe	Выполнено установление данных
VMA	Valid Executive Data	Адрес памяти установлен
VPA	Valid Memory Address	Адрес внешнего устройства установлен
VVA	Valid Periferi Address	Адрес пользователя установлен
VXA	Valid Vser's Address	Установлено выполнение адресации
W, WR	Valid Executive Address	Запись
WAITRQ	Write	Запрос при ожидании
W/B	Wait Request	Работа словами/байтами
WE	Word/Byte	Разрешение записи
W/LW	Write Enable	Работа словами/удлиненными словами
WO	Word/Long-word	Запись в вывод
WT	Write-Output	Ожидание
WTBT	Wait	Запись байтами
XA	Write Byte	Линии арбитража
XACK	Arbitration	Подтверждение передачи данных
XFRACK	Transfer Acknowledge	Подтверждение быстрого запроса
	Fast Request Acknowledge	

УДК 681.325

А. П. Дианов, Н. Н. Щелкунов

МОДУЛИ ПРОГРАММИРОВАНИЯ ЛОГИЧЕСКИХ СХЕМ

(Продолжение цикла. Начало см. №№ 5, 6, 1986 г., №№ 1—5, 1987 г.)

Запись информации в ПЗУ и матричные интегральные схемы являются важнейшей операцией в процессе разработки, производства и эксплуатации микропроцессорных средств и систем. Для поддержки этой операции служат различного рода программирующие устройства и системы (программаторы), которые выполняются как в автономном, так и во встроенном в системы проектирования вариантах.

В литературе можно найти описание нескольких типов программаторов ПЗУ [1, 2] и матричной логики [3, 4]. Все эти средства ориентированы на непосредственное подключение к системной магистрали базовой микроЭВМ и на отдельные классы программируемых приборов. Недостатки программаторов во многом были предопределены ориентацией на системные интерфейсы общего назначения, никак не учитывающие специфические потребности модулей.

Для системы проектирования, имеющей встроенный интерфейс МИ8102 [5, 6], задача проектирования аппаратуры программаторов сводится к разработке набора модулей программирования. Основное назначение модулей состоит в конкретизации функционального назначения линий МИ8102 применительно к выбранному типу программируемого прибора, геометрии его корпуса и физической разводке выводов. Многие типы подобных приборов могут быть сведены в один класс, для которого разрабатывается единый модуль программирования.

Предлагается набор из семи модулей:

Модуль	Микросхемы, обслуживаемые модулем
ММС9411	К573РФ2, К573РФ5
ММС9412	К573РФ4, К573РФ6, 2764, 2704А, 2704L, 27128, 27128А, 27259, 27259D, К558РФ3
ММС9413	27512
ММС9421	К556РТ4, К556РТ5, К155РЕ3
ММС9422	К556РТ6, К556РТ7
ММС9423	К556РТ1, К556РТ2
ММС9424	К556РТ11, К556РТ12, К556РТ13, К556РТ14, К556РТ15, К556РТ16, К556РТ17, К556РТ18

Модули стыкуются непосредственно с интерфейсом проектирования, формируемым адаптером ММС8102, что обеспечивает простоту их принципиальных схем. Адаптер ММС8102 является обязательным элементом системы проектирования, который реализует функцию универсального интерфейса проектирования с широким спектром приложений.

Основу программного обеспечения системы составляет оверлейный пакет с головной программой МРРS, работающий в операционной среде ДОС1800. Основная особенность пакета состоит в использовании ряда фай-

лов, соответствующих отдельным микросхемам, модулям программирования или их объединениям. Расширение множества обслуживаемых программатором микросхем, а также алгоритмов их программирования зависит только от написания новых и модификации старых файлов в рамках ряда требований, накладываемых на них. Эти требования составлены таким образом, чтобы максимально облегчить процесс работы с файлами.

Модули серии ММС941Х предназначены для записи информации в перепрограммируемые ПЗУ с ультрафиолетовым (серии К573, 2700) либо электрическим (серия К558) стиранием. Методика их программирования довольно простая [7, 8] и реализуется в простом аппаратном интерфейсе для записи данных. Микросхемы могут непосредственно подключаться к интерфейсу проектирования типа МИ8102, а роль модуля программирования сводится к физической стыковке интерфейса с конкретным прибором. При реализации процедур программирования микросхем определенного класса канал А1 интерфейса проектирования используется для ввода-вывода данных, каналы В1 и С1 — для вывода адреса, линии канала С2 — для формирования сигналов управления прибором. При этом источник Е1 резервируется для управления напряжением питания U_{cc} , а Е2 — для контролируемого программой изменения напряжения программирования U_{pp} .

Разводка модулей ММС9411 и ММС9412 (рис. 1,2) формировалась постепенно по мере расширения состава обслуживаемых микросхем. В процессе работы довольно легко спроектировать другие более удачные разводки персональных модулей для данного класса приборов, однако при этом придется перерабатывать связанные с ними и уже отлаженные персональные файлы.

Для формирования таких сигналов управления приборами ПЗУ как $\sim CE$, $\sim OE$, $\sim E_p$, $\sim PGM$ служат линии С25...С27, что обеспечивает ряд удобств при реализации адаптивных алгоритмов программирования [7], которые применяются для большинства типов микросхем. Действительно, при требуемом постоянном перепрограммировании канала А1 из режима вывода данных в режим ввода и наоборот состояние линий С2, а

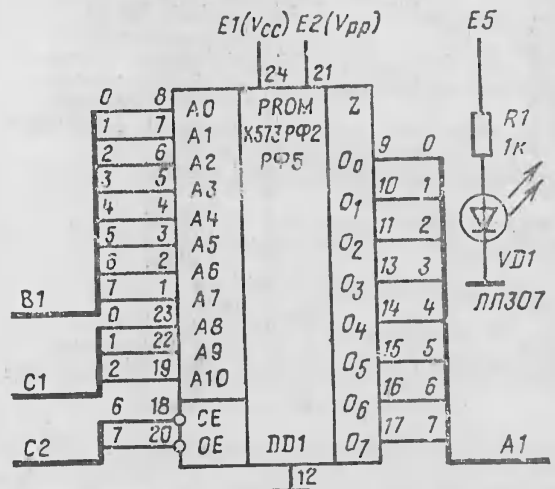


Рис. 1. Схема модуля ММС9411

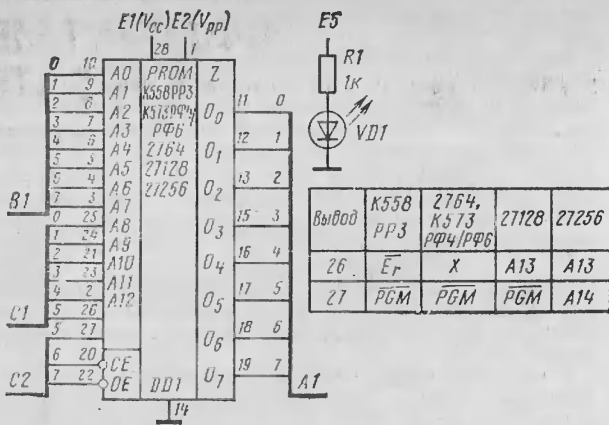


Рис. 2. Схема модуля MMC9412

следовательно, и управляющих сигналов не изменяется, что очень важно для самого алгоритма.

Светодиод VD1 играет вспомогательную роль. Он используется для индикации текущего рабочего состояния программатора, информирующего пользователя о возможности устанавливать или вынимать очередную микросхему ПЗУ из панельки персонального модуля. Один из естественных вариантов его использования состоит в индикации состояния, когда все подаваемые на панельку напряжения сняты.

Модуль MMC9413 является дальнейшим развитием MMC9412 (рис. 3). В нем предусмотрена возможность считывания программатором типа прибора и кода его изготовителя, которая реализуется для ряда схем в специальном режиме идентификации [9]. Для активизации данного режима необходимо на адресную линию A9 прибора подать напряжение $12 \pm 0,5$ В. После чего два байтовых идентификатора могут быть считаны с выходов микросхемы, адреса которых осуществляется с помощью линии A0. При этом все другие управляющие и адресные линии должны быть удержаны в состоянии низкого напряжения. Код изготовителя считывается при $A0=0$, код типа прибора — при $A0=1$. Примеры этих кодов для ряда микросхем фирмы даны ниже:

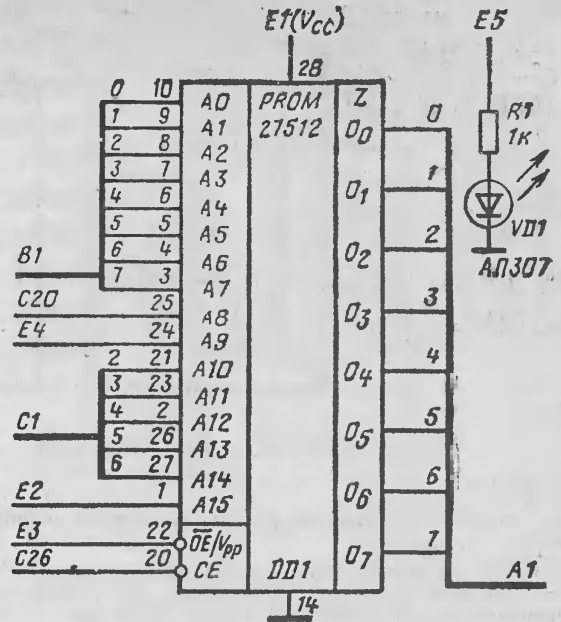


Рис. 3. Схема модуля MMC9413

Микросхема	Код изготовителя	Код типа ИС
2732A	89H	01H
2764A	89H	08H
27C64	89H	07H
87C64	89H	37H
27128	89H	83H
27128A	89H	89H
27256	89H	04H
27512	89H	0DH

Восьмой разряд всех кодов содержит признак контроля на четность.

Использованная в модуле MMC9413 универсальная разводка дает возможность постепенно расширить множество обслуживаемых типов микросхем. Введение процедуры автоматической идентификации типа прибора предохранит пользователя от ряда потенциальных оши-

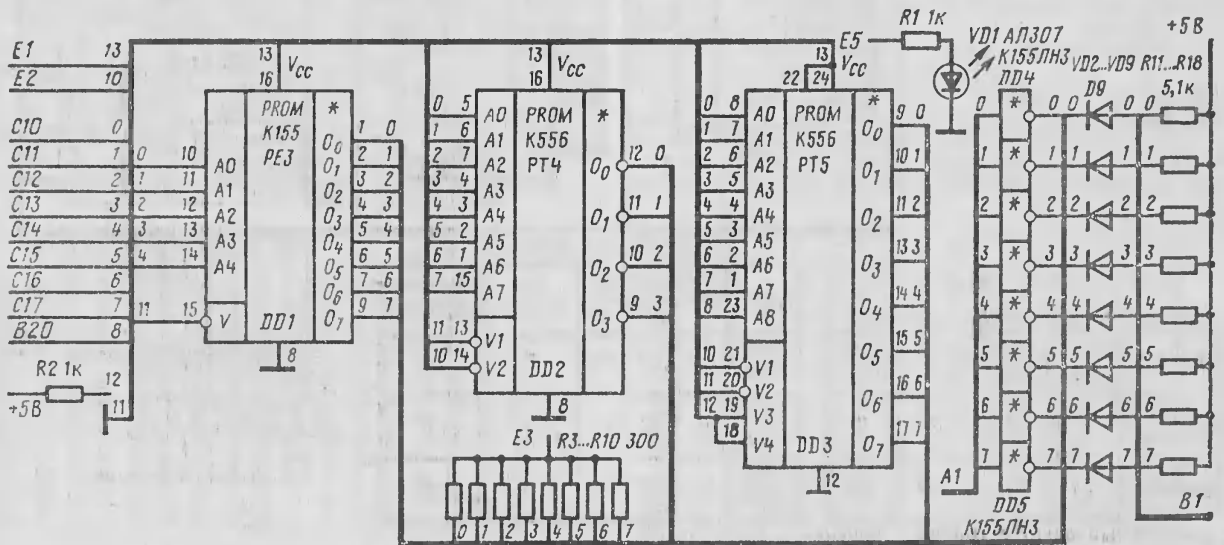


Рис. 4. Схема модуля MMC9421

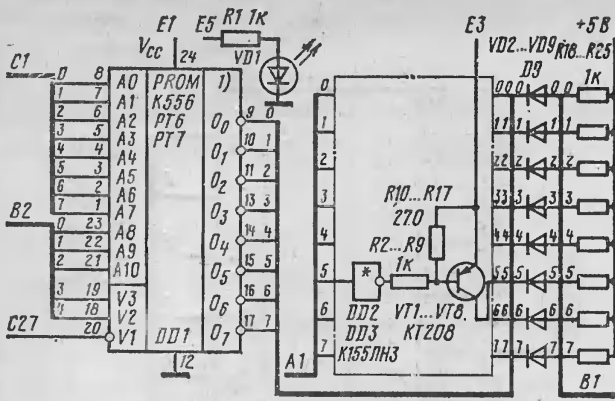


Рис. 5. Схема модуля мМС9422

ния амплитуды импульса прожига на одном из выходов ПЗУ.

Модули мМС9421/9424 имеют по три панельки для установки микросхем, что объясняется существенными различиями в функциональном назначении одних и тех же линий физического интерфейса для разных приборов. При программировании используется одна из панелек, другие при этом должны быть свободны от микросхем. Для формирования программирующих импульсов требуемой амплитуды на выходах микросхем предусмотрены простые ключи, собранные на инверторах К155ЛН3 (рис. 4) с высоковольтным выходом типа открытого коллектора и транзисторах КТ208 (рис. 5,6). Германиевые диоды D9 защищают входы канала В1 от прохождения на них импульсов программирования, имеющую амплитуду Е3.

В последнее время в составе серии К556 появился ряд новых элементов с упрощенной техникой программирования. Для их прожига уже не требуется источник Е2. Эти микросхемы обслуживаются отдельным персональным модулем мМС9424 (см. рис. 6).

Персональный модуль мМС9423 для программирования ПЛМ. Наибольшую сложность для разработчика программирующих систем представляют модули для записи информации в программируемые логические матрицы, что объясняется сравнительной сложностью методики их программирования [10]. На рис. 7 представлена схема персонального модуля мМС9423, предназначенного для обслуживания ПЛМ серии К556. Схема модуля учитывает опыт проектирования и эксплуатации системы программирования логических приборов [4].

Адресные ключи К1...К16 собраны с использованием микросхем К564ЛА7 и К543КН3. Управление работой

блок, которые могут возникнуть при физических манипуляциях с прибором.

Модули серии мМС942Х для программирования ПЗУ предназначены для занесения информации в однократно программируемые ПЗУ серий К155 и К556 (рис. 4—6). В модулях данной серии канал А1 используется для вывода, а канал В1— для ввода данных. Адрес выводится через шины С1 и В2. Канал С2 служит для формирования управляющих сигналов. Для управления напряжением питания Ucc используется источник Е1, для управления напряжением на одном из входов V—Е2. Третий источник Е3 необходим для программирова-

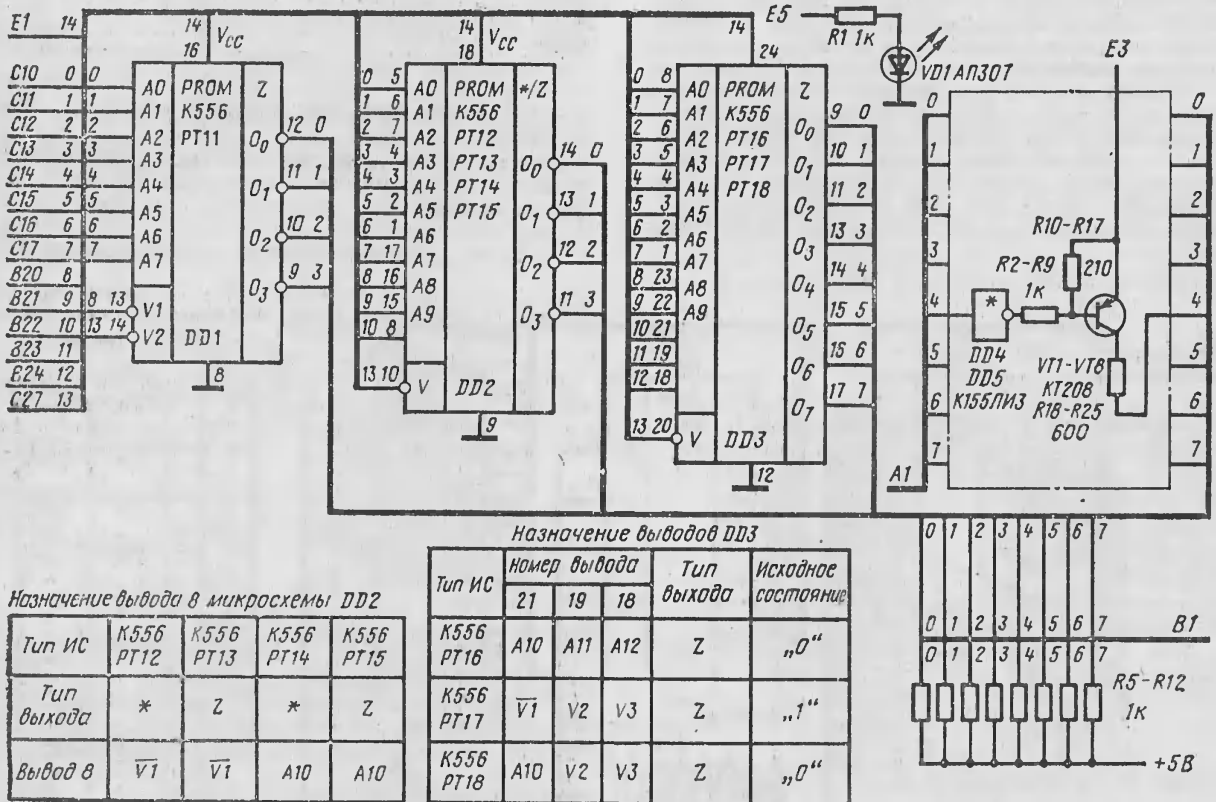


Рис. 6. Схема модуля мМС9424

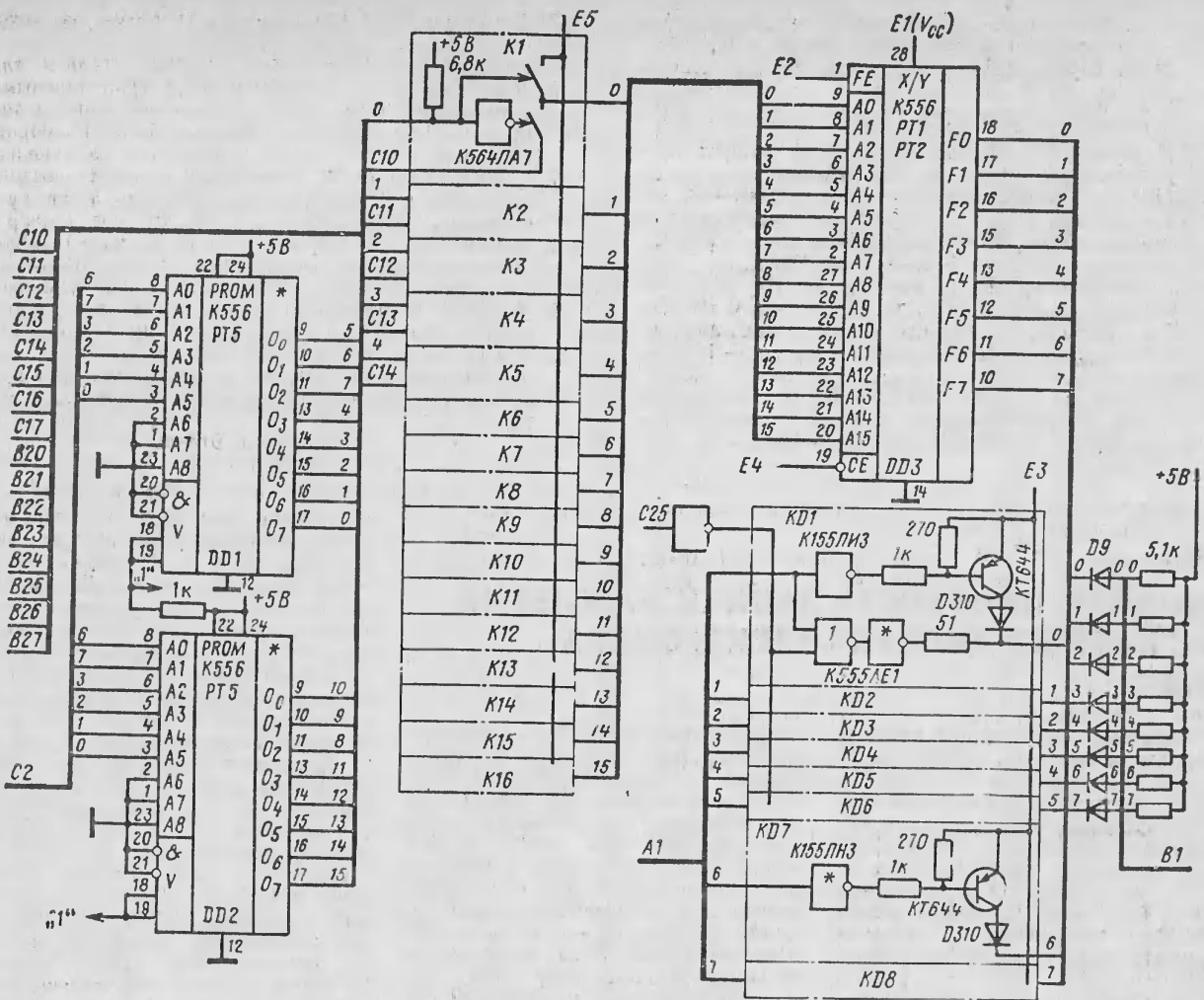


Рис. 7. Схема модуля мМС9423

осуществляется через дешифратор, реализованный на двух ПЗУ К556РТ5 DD1, DD2, карта прожига которых приведена на рис. 8, 9 соответственно. Алгоритм работы устройства следующий. При C26=C27 на адресных входах DD3 присутствует код шины C1 и B2 в ТТЛ-уровнях. При C26=1, C27=0 на входы прикладной ПЛМ DD3 подается напряжение E5=10 В. В последнем случае при C26=0, C27=1 на адресный вход ПЛМ, номер которого определяется кодом C23...C20,

подается ТТЛ-уровень соответствующей линии выходной шины C1, B2, а на все остальные адресные входы — напряжение E5. Это важно для реализации процедур программирования слоя И.

Ключи данных KD1...KD8, построенные на транзисторах KT644, позволяют при C25=1 формировать импульсы напряжения с перепадом от ТТЛ-уровня «Лог. 1» и до величины E3. Такой режим работы необходим для программирования как слоя ИЛИ, так и слоя НЕ. При

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.000.	00	FF	FF	00	00	FF	FB	00	00	FF	FF	00	00	FF	EF	00
.010.	00	FF	FF	00	00	FF	FE	00	00	FF	FF	00	00	FF	BF	00
.020.	00	FF	FF	00	00	FF	FD	00	00	FF	FF	00	00	FF	DF	00
.030.	00	FF	FF	00	00	FF	F7	00	00	FF	FF	00	00	FF	7F	00
.040.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.050.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.060.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.070.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.080.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.090.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0A0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0B0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0C0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0D0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0E0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0F0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Рис. 8. Карта прожига микросхемы DD1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.00.	00	FF	7F	00	00	FF	FF	00	00	FF	F7	00	00	FF	FF	00
.010.	00	FF	DF	00	00	FF	FF	00	00	FF	FD	00	00	FF	FF	00
.020.	00	FF	BF	00	00	FF	FF	00	00	FF	FE	00	00	FF	FF	00
.030.	00	FF	EF	00	00	FF	FF	00	00	FF	FB	00	00	FF	FF	00
.040.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.050.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.060.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.070.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.080.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.090.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0A0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0B0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0C0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0D0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0E0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
.0F0.	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Рис. 9. Карта прожига микросхемы DD2

C25=0 на выводах F0...F5 формируется двоичный код с уровнями, равными напряжению «Лог. 0» и ЕЗ, который обычно служит для адресации конъюнктора матрицы И.

ЛИТЕРАТУРА

1. Дианов А. П., Щелкунов Н. Н. Модуль программирования микросхем ПЗУ//Микропроцессорные средства и системы. — 1985. — № 3. — С. 80—83.
2. Лукьянов Д. А. Схематехника универсальных программаторов ПЗУ//Микропроцессорные средства и системы. — 1985. — № 3. — С. 84—88.
3. Корсаков С. Я., Крылов В. В., Кочетков А. А., Морозов А. В., Прошин В. И. Система проектирования программируемой логики//Микропроцессорные средства и системы. — 1985. — № 2. — С. 40—44.
4. Дианов А. П., Щелкунов Н. Н. Технические средства программирования логических схем//Микропроцессорные средства и системы. — 1986. — № 2. — С. 77—80.

УДК 681.323

В. Б. Кочубин

ДВОЙНАЯ ПЛОТНОСТЬ ЗАПИСИ НА ГИБКИХ МАГНИТНЫХ ДИСКАХ «ЭЛЕКТРОНИКА ГМД-7012»

Режим двойной плотности реализуется при совместной работе контроллера с интерфейсом МС 4701. Работа с указанными устройствами поддерживается драйвером «DX» в ОС РАФОС II. Подготовка аппаратной части сводится к удалению перемычки Е1 на плате КЗ контроллера НГМД (при необходимости можно установить переключатель, зашунтировав его конденсатором не менее 3 нФ).

В регистр состояния интерфейса (177170) заносится коды функций: установка плотности записи, номер привода, пуск 1 (для приводов 0—11 и 1—31 восьмеричное). Затем в регистр данных (177172) записывается 1 для двойной плотности или 0 для одинарной. После завершения этой

операции диск может быть инициализирован только драйвером соответствующей плотности (программа DUP, которая вызывается для инициализации, обращается к устройству через драйвер). Для того чтобы НГМД можно было использовать в качестве системного устройства, необходимо изменить программу аппаратной загрузки, которая размещается в ПЗУ с адреса 173000 (ПЗУ расположено на плате И4), и драйвер «DX». Алгоритмы программ управления НГМД начинаются с блока «заслать команду», затем идет блок «задание плотности». Предлагается ввести во все алгоритмы после указанных блоков два дополнительных блока: «задние

```

* ENABL LC
* АППАРАТНЫЙ ЗАГРУЗЧИК ГИБКИХ ДИСКОВ *
* АППАРАТНЫЙ ЗАГРУЗЧИК ГИБКИХ ДИСКОВ *

* TITLE АППАРАТНЫЙ ЗАГРУЗЧИК DX
* IDENT /04.01/
* IIF NDF FOR00, FOR00 =0
* IIF NDF FOR01, FOR01 =0
* IIF NDF FOR10, FOR10 =0
* IIF NDF FOR11, FOR11 =0
* IIF NDF STARTDX, STARTDX=173000

* STARTDX
MOV #100247, R2
MOV #177170, R1
AAA: BIT R2, R1
BEQ AAA
MOV #7, R3
MOV R1, R0
MOV R2, (R0)+

* IF NE FOR00!FOR01!FOR10!FOR11
POP: TSTB BR1
EPL POP
* IF NE FOR00!FOR01
MOV #0, ER0
* ENDC
* IF NE FOR10!FOR11
MOV #1, ER0
* ENDC
* ENDC

```

```

BR #0, ER0
MOV ASR R3
BCS CCC
MOV #11023, R1
* IF NE FOR00!FOR01!FOR10!FOR11
PO: TSTB R1
BPL PO
* IF NE FOR00!FOR01
MOV #0, ER0
* ENDC
* IF NE FOR10!FOR11
MOV #1, ER0
* ENDC
* ENDC
CCC: BIT R2, R1
BEQ CCC
BMI DDD
BCS EEE
TSTB R1
* IF NE FOR00!FOR01!FOR10!FOR11
BMI G11
* IFF
* WORD 100771
* ENDC
CLR R0
CMP #240, ER0
BNE DDD
CMPB #247, R2
ADC R0
CLR PC

```

Рис. 1. Программа аппаратной загрузки

5. Дианов А. П., Щелкунов Н. Н. Универсальный одноплатный микроконтроллер//Микропроцессорные средства и системы. — 1986. — № 5. — С. 65.
6. Дианов А. П., Щелкунов Н. Н. Система проектирования микропроцессорных устройств//Микропроцессорные средства и системы. — 1987. — № 5. — С. 83—86.
7. Щербаков О. А. Физические основы записи информации в ПЗУ//Микропроцессорные средства и системы. — 1985. — № 3. — С. 72—75.
8. Дианов А. П., Щелкунов Н. Н. Методика программирования микросхем ПЗУ//Микропроцессорные средства и системы. — 1985. — № 3. — С. 75—79.
9. Memory Components Handbook (Order Number 210830—004) // Intel Corp., USA, 1985.
10. Дианов А. П., Щелкунов Н. Н. Процедуры программирования логических матриц//Микропроцессорные средства и системы. — 1986. — № 2. — С. 71—76.

Статья поступила 11 ноября 1986 г.

плотности» (при этом информация о плотности передается 00 разрядом регистра данных) и «ожидание запроса передачи». Последующие блоки алгоритмов выполняются без изменений. Контроллер начинает работать по новому протоколу обмена с удалением перемычки Е1. Новая программа аппаратной загрузки после ассемблирования занимает 200 восьмеричных ячеек памяти, что в два раза превышает объем, который обеспечивается штатным ПЗУ на плате И4 (рис. 1). Для размещения четырех корпусов ПЗУ К155РЕЗ необходимо доработать плату И4.

Программа АЗ может быть размещена в ОЗУ и загружаться системой, записанной на НГМД с одинарной плотностью. Затем перемычка Е1 отсоединяется, устанавливается НГМД с двойной плотностью с системой и драйвером двойной плотности. Запуск АЗ производится без отключения питания. При объеме ОЗУ 56К байт, АЗ может разместиться с адреса 100000 (восьмеричное).

Подготовка программной части сводится к трансляции исходного текста

```

* IF NE FOR00!FOR01!FOR10!FOR11
PNT1:: MOV #3, R4
IOT
* IF NE FOR10!FOR11
MOV #1, ER5
* IFF
MOV #0, ER5
* ENDC
RTS PC
PNT0:: IOT
* IF NE FOR10!FOR11
MOV #1, ER5
* IFF
MOV #0, ER5
* ENDC
IOT
MOV R3, ER5
RTS PC
* IF NE FOR10!FOR11
PNT2:: TST R4
BMI B10
RTS PC
MOV #40000, R4
TST R4
BEQ G00
MOV #120, 2(SP)
RTS PC
* ENDC
* ENDC
G11: MOV ER0, (R3)+
BR CCC
* END ST

```

```

ENABL DD
TITLE DX ОДИНАРНОГО И ДВОЙНОГО ПЛОТНОСТИ.
IDENT 704.01
SBTTL RACS FLOPPY DISK HANDLER
ELDX == 1
MACROS AND DEFINITIONS
MCALL CRDEF
MACRO .ASSUME A1,END,A2
CNQ (A1)-(A2)
IF
IFB
ERROR
ENDC
ENJM .ASSUME
MACRO .BR TO
IF DF TO
IF NE -(TO)
ERROR
ENDC
ENDC
ENJM .BY
MDEF DX,22,FILSTX,SPFUNK,DXWCD,DDSSBL,
17770,264
IF NDF DXNO,DXNO=0
IF NDF FOR00,FOR00=0 ;DA,PA
IF NDF FOR0,FOR0=0 ;DA,PA,DX0,AB,PA,DX1
IF NDF FOR10,FOR10=0 ;AB,PA,DX0,DA,PA,DX1
IF NDF FOR1,FOR1=0 ;AB,PA
IF NE FOR01,FOR10,FOR10,FOR11,FOR=0 ;PARAMETP
ПРОТОКОЛА ОБМЕНА
IF NDF FOR0,FOR=0 ;PARAMETP СВОЕГО ДИСКА
IF NE FOR0,FOR=1
IF NDF STARTDX,STARTDX = 364000;CL,PA,А.ЗАПЫЧКА
FORBLK = 0
IF NE FOR11,FOR01,FORBLK = 1
FOR = 0
IF NE FOR01,FOR01,FOR01,FOR11,FOR=1
IF EQ (FOR0-1)
DDSSBL =494,*2
IF EQ (FOR0-1)
DDSSBL=DDSSBL*2
ENDC
DXNREG = 3
RETRY = 1C
SPFUNK = 100000
USW = 44
IF NDF DXWC2,DXWC2 = 270
IF NDF DXWCS2,DXWCS2 = 477474
ESGO = 1
ESUNIT = 20
ESDONE = 40
ESINT = 100
ESIR = 200
ESIN = 400
ESR02 = 4000
ESINIT = 40000
ESERR = 100000
ESBUF = 0*2
ESBUF = 1*2
ESURT = 2*2
ESRD = 3*2
ESRUST = 0*2
ESURTD = 0*2
ESMAIN = 7*2
ASSUME CSRD2 NE 0
ASSUME CSURD2 EQ 0
ASSUME CSURD2 EQ 0
DNERR = 20
ESGRS = 2
ESPAR = 2
ESID = 4
ESSIDI = 10
ESDNER = 20
ESDN = 40
ESDD = 100
ESDRY = 200
IF NE MPMGT
IF EQ TSGM
PAR = 172342
IF
PAR = 172354
ENDC
SBTTL INSTALLATION CHECK9
ASECT
MOP
BIT #CSR02,0176
BNE 1X
TST (PC)+
3X:
SEC
RTS PC
SBTTL DRIVER REQUEST ENTRY POINT
ENABL
DRBEG DX
MOU #RTRV,(PC)+
RTRV:
WORD 0
MOU DXCDE,R3
MOU (R3)+,R5
MOU #CSR01CSG0,R4
MOUB (R3)+,R1
MOUB (R3)+,R0
IF EQ FOR CLR (PC)+
UNIT:
WORD
ENDC
IF NE FOR01,FOR01
MOU #1,(PC)+
UNIT:
WORD
ENDC
IF NE FOR01,FOR00
CLR (PC)+
UNIT:
WORD
ENDC
ASRB R0
BCC 1X
BIS #CSUNIT,R4
IF NE FORW JMP RXERR
ENDC
IF NE FOR10 CLR UNIT
ENDC
IF NE FOR01
MOU #1,UNIT
IF EQ DXNO BIT #C2,R0
BCC SRXERR
JMP
SRXERR:
IF
MOU #DXCSR,RXCSA
R0
BCC 2X
MOU #DXWCS2,RXCSA
R0
BCC SRXERR
ENDC
MOU (R3)+,R0
MOU (R3)+,R2
BPL 3X
CMPB -(R4)+,(R4)
ASSUME CSURT EQ CSRD-2
NEG R2
3X:
ADD R0,R1
ADD DXGTBL,-(R1),R4
MOU MOUB (PC)+,R4
BNI 7X
MOUB DECB FILLCT
R2,FILLCT
ASL R2
BIT #1,UNIT
BNE N2
ASL R5
ASL R5
MOU (PC)+,R4
BYTE -7,-26.
4X:
CMP #26,#200,R5
BHI 5X
ADD #26,#200,R5
ROL R5
JNCB 4X
BLE 4X
MOUB R5,R1
R5,R1
ADD R4,R5
MOU R1,R4
ASL R1
ADD R4,R1
ASL R1
SUB #26,R1
BGT 7X
MOU MOU R1,TRKOFF
BR 8X
7X:
SWAB R5
BISB R2,R5
BIT #1,UNIT
BEQ N1
MOU #400,R2
BR N01
N1:
MOU #120,R2
IF EQ MMGT CLR (R0)+
IF
MOU DXCDE,R4
CLR -CS
JSR PC,#PTRD
TST (R0)+
ENDC
MOU R0,(PC)+
B:
BUFRAD:
WORD 0
IF NE FORW R5,TRACK
CHP TRACK*#76.
BLE LOON
SUB #476,R5
MOU #26,TRKOFF
BIS #CSUNIT,R4
BIS #CSUNIT,RXFUN2
LOON:
ENDC
MOU R5,TRACK
MOU MOU R2,(PC)+
BYTNT:
WORD 0
IF NE MMGT
TST (R3)+
TST #0R3,(PC)+
PARUAL:
WORD 0
ENDC
BR RXINIT
DSABL LSB
SBTTL START TRANSFER OR RETRY
ENABL:
MOU #100000,RXIRTN
RAXINIT:
MOU RXCSA,R4
BR RXIENB
1X:
BIT #2,R0
BNE 3X
2X:
G12:
WORD
MOUB R0,SILOFE
CSBUF,CSG0
(R2)+,R5
3X:
MOU MOUB R1,R5
BGT BGT 5X
SUB #14,R2
BGT 4X
ADD #12,R2
4X:
ROL R2
ADD (PC)+,R2
TRKOFF:
WORD 0
BGT 5X
ADD #26,R2
5X:
MOU MOU R0,OR4
JSR PC,WAIT
6X:
TSTB BR4
BFC BFC
BNI BRTRY
NAXRTV:
MOUB R2,OR5
TSTB BR4
BEQ 7X
RAXTRY:
(PC)+,OR5
TRACK:
BYTE 0
SECTOR:
BYTE 0
RXIENB:
BIS #CSINT,OR4
PC
RTS
RXERR:
MOU DXCDE,R4
BIS #DERR,0-(R4)
B:
DRAST DX,5,RXABRT
FDRK DX,BLK
RAXFUN2:
MOU (PC)+,R0
WORD 0
BEQ #1,UNIT
MOU #400,R3
BR N03
N03:
MOU #120,R3
RAXCSA:
WORD
MOU DXCSR,R4
R4,R5
TST (R5)+
RAXTRY:
(PC)+
RAXRTN:
WORD 0
BCC 1X
BIT #2,R0
TST R0
BPL 9X
BIT #ESDD,OR5
BGT 9X
IF EQ BUFRAD:
WORD 0
INC BUFRAD,R2
IF
MOU R4,R1
MOU DXCDE,R4
MOU #1,-CS
MOU #2,0,BUFF-0,BLN(R4)
JSR PC,#PTRD
R1,R4
ENDC
9X:
JSR R0,SILOFE
CSBUF,CSG0
OR5,(R2)+
OR5,R2
INCB SECTOR
11X:
BNE 11X
ADD #26,#400+1,TRACK
IF NE FORW TRACK,CHP
BLT COR
MOU #26,TRKOFF
SUB #76,TRACK
BIS #CSUNIT,RXFUN2
BIS #CSUNIT,R0
11X:
ENDC
COR:
ADD #6,TRKOFF
BLE 11X
SUB #26,TRKOFF
BR 11X
11X:
IF EQ MMGT ADD R3,BUFRAD
IF
ADD #2,PARUAL
ENDC
SUB R3,BYTNT
BNI CLR
CLR BYTNT
BIT #2,SPFUNK,R0
BNE 12X
BIT #1,UNIT
BGT BGD
ADD #100000,FILIN
BR BCC
BDD:
ADD #0*4000,(PC)+
FILIN:
BYTE 0
FILLCT:
BYTE 0
BCCA:
BCC 12X
JMP SI2
12X:
IF NE ERLNG
MOU #DXCOD*400+377,R4
DXCDE,R5
JSR PC,#ELPTR
ENDC
13X:
CLR @RXCSA
14X:
JDFIN
RAXRBT:
MOU #CSINIT,ORCSA
CLR DXFLK+2
BR 14X
DSABL LSB
DXNO BIT #C2,R0
BCC SRXERR
ENDC
RAXTRY:
IF ERLNG
MOU #DXRBUF-,R3
PC,R3
ADD R3,R2
MOU OR5,(R3)+
MOU OR5,(R3)+
MOU #CSMAIN,CSG0,OR4
PC,WAIT
1X:
BIT #CSDONE,OR4
BGT 1X
MOU OR5,OR3
MOU #RETRY*400+DXNREG,R3
MOU #DXCOD*400,R4
MOU R3,R4
MOU DXCDE,R5
JSR PC,#ELPTR
RAXCSA,R4
ENDC
DEC RAXTRY
IF NE MMGT & DXNO
BGT 2X
JMP RXERR
2X:
IF BGT
JMP CRXERR
CRXERR:
ENDC
MOU #CSINIT,OR4
ADD #2,R4
MOU UNIT,OR4
MOU #2,R4
JMP RAXINIT
SBTTL SILOFE - FILL OR EMPTY THE SILO
WAITI:
IF NE FOR
TIT:
EQ OR4
EQ TIT
ADD #2,R4
MOU UNIT,OR4
SUB #2,R4
ENDC
RTS PC

```

Рис. 2. Текст драйвера DX (начало)

```

SLOPE: MOV <R0>+,OR6
        JSR PC,WAIT
        MOV <R0>+,S4
        MOV <R0>+,S4
        MOV BVICNT,R1
        BEQ 4x
        .IF NE FPGAT
        MOV BAPAR-(SP)
        MOV PARPAR,CMPAR
        .ENDC
        CMP R1,R3
        BLOS 1x
        MOV R3,R1
        MOV BNEPAR,R2
        1x: TSTB OR4
        2x: BPL 2x
        3x: HALT
        TSTB OR4
        JCRB R1
        BNE 2x
        .IF NE FPGAT
        MOV <SP>+,CMPAR
        .ENDC
        4x: TSTB OR4
        BEQ 5x
        BEQ 4x
        5x: HALT
        BR 4x
        6x: RIS
        .SBITL INTEL: FOMK BLOCX, END OF DRIVER
        .WORD CSWRD+CSRD+SPFUNG
        .WORD CSWRD+CSRD+SPFUNG
        .WORD CSWRD+CSRD+SPFUNG
        .WORD 0
        .DIBLK: .WORD 0,0,0,0
        .IF NE ERJNG
        .DIBLK: .DIBLK
        .ENDC
        .SBITL BOOTSTRAP DRIVER
        .DIBOT DX,BOOT,READ
        = DXBOOT+14
        .WORD READS-DXBOOT
        .WORD 340
        .WORD WAIT-DXBOOT
        .WORD 340
        = DXBOOT+34
        .DIBOT: MOV UNTRD-DXBOOT(R0),RDCMD
        REETRY: MOV BPT,SP
        .IF NE FOR10:FOR11
        MOV #400,R2
        .IFB
        MOV #200,R2
        .ENDC
        CLR R0
        BR 2x
        = DXBOOT+56
        .DIBOT: .BYTE CS60+CSRD
        .IF NE FOR4
        .BYTE CS60+CSRD
        .IFB
        .BYTE CS60+CSRD+CSUNIT,READ FROM UNIT 1
        .ENDC
        = DXBOOT+70
        WAIT: TST OR4
        BEQ BPT
        REETRY
        RTIRET: RIL
        = DXBOOT+120
        READS: MOV #DIBESR,R4
        MOV R4,R5
        MOV <PC>+,<R5>+
        B 0
        RDCMD: .WORD 0
        .IF NE FOR
        JSR PC,@<STARTDX+132>
        .IFB
        JOT MOV R3,OR5
        .ENDC
        JOT MOV R0,OR5
        JOT
        .IF NE FOR
        JSR PC,@<STARTDX+116>
        .IFB
        MOV #CS60+CSEBUF,OR4;LOAD EMPTY
        ; BUFFER FUNCTION
        ; INTO RDCSR
        .ENDC
        EROFES = READF
        RDX: JOT TSTB OR4
        BPL RTIRET
        MOV OR5,<R2>+
        DEC R1
        BEQ RDX
        CLR R2
        BR RDX
        BR SP,R1
        INC R0
        MOV GPC,R3
        BPT
        BOOT2: .IF EQ FOR10:FOR11
        MOV <R3>+,<R3>+
        BPT
        MOV <R3>+,<R3>+
        BPT
        .IFB
        NOP
        NOP
        NOP
        NOP
        .ENDC
        BIT #CSUNIT,RDCMD
        BNE BOOT
        CLR R0
        BR BOOT
        MOV <PC>+,<PC>+
        .WORD 167
        .WORD RDX-DXBOOT1
        .WORD <PC>+,<PC>+
        .WORD READF-RDX-4
        .WORD RDX-DXBOOT+2
        .WORD READ1-DXBOT,@BNEREAD
        MOV #TRWAIT-DXBOT,G#20
        CLR OR4
        HRD801
        TST HRD801
        BEQ READ1
        MOV UNTRD-DXBOOT(R3),RDCMD
        FOR10:FOR11
        JIF NE FOR10:FOR11
        NOP
        .IFB
        ASL R0
        MOV R0,R3
        CLR R0
        BR 3x
        SUB #23,R3
        INC #25,R4
        SUB 2x
        CMP #14,R4
        RCL R3
        SUB #25,R3
        4x: BPL ADD
        ADD #27,R3
        BPT
        MOV <SP>+,R0
        INC R0
        TST R1
        BGT 1x
        RETURN
        TST OR4
        BEQ READF
        BMT BICERR
        TSTB OR4
        BPL READFX
        MOV OR5,<R2>+
        DEC R1
        BEQ READF
        MOV R1,R2
        READF
        TRWAIT: .IF NE FOR10:FOR11
        JSR PC,@<STARTDX+146>
        .IFB
        TST OR4
        BMT BICERR
        .ENDC
        BEQ TRWAIT
        READFX: RTI
        = DXBOOT+686
        BOOT: MOV #10000,SP
        MOV R0,<SP>
        MOV #2,R0
        MOV #4400,R1
        MOV #1000,R2
        CLR <PC>+
        .WORD 1
        HRD801: JSR PC,READ
        MOV #READ1-DXBOT,@BNEREAD
        MOV #BNDNAM,@BNEDEUN
        MOV <SP>+,@BNEDEUN
        JMP @BNEBOOT
        .ENDC
        .JRENO DX
    
```

Рис. 2. Текст драйвера DX (окончание)

драйвера DX (рис. 2) совместно с файлами параметров генерации драйвера (CND.MAC) и параметров генерации монитора (SJCND.MAC). Файл SJCND.MAC создается в процессе генерации монитора. Если трансляция осуществляется без CND.MAC, то генерируется стандартный драйвер DX. Для получения драйвера, работающего с двумя приводами с двойной плотностью, необходимо определить параметр FOR11=1 и адрес, с которого будет размещаться A3: STARTDX=10000. Последнее определение задавать необходимо, так как драйвер при загрузке обращается к инструкциям, размещенным в A3. В ОС РАФОС II операция замены драйвера системного устройства не требует регенерации монитора, поэтому на системном устройстве можно размещать несколько специальных драйверов DX с измененной спецификацией и по мере необходимости подключать соответствующий драйвер, переименовывая его в DXS.SYS. На системном ГМД с двойной плотностью целесообразно размещать драйверы для работы с приводом 0 — двойная плотность, приводом 1 — одинарная плотность (параметр FOR10=1), приводами 0 и 1 — двойная плотность (параметр FOR11=1) и драйвер, объединяющий оба привода при двойной плот-

ности (параметры: FOR11=1 и FORW=1). Для генерации драйвера, работающего с одинарной плотностью с приводами 0 и 1, необходимо определить FOR00=1, а для работы с приводом 0 при одинарной плотности и приводом 1 при двойной плотности — FOR01=1. Параметр FORW=1 совместно с параметром FOR00=1 вызывает генерацию драйвера, объединяющего приводы 0 и 1 с одинарной плотностью. Файл параметров генерации может выглядеть так:

```

;FOR00=1
;FOR01=1
;FOR10=1
;FOR11=1
;FORW=1
STARTDX=10000
    
```

При этом будет сгенерирован драйвер, работающий с двойной плотностью с приводом 1, и приводом 0 с одинарной. Этот драйвер необходим для предварительного копирования файлов на ГМД с двойной плотностью на комплексах, имеющих одно дисковое устройство. Командный файл трансляции драйвера записывается следующим образом:

```

.ASS DXI DK
MACRO/OBJ: DXS CND+SJCND+
+DX
.LINK/EXE: DXS.SYS DXS
.DEL/NOQ DXS.OBJ
    
```

Командный файл трансляции A3 имеет следующий вид:

```

.ASS DXI DK
MACRO/OBJ: AV CND+AV
.LINK/BOTTOM: n AV
.DEL/NOQ AV.OBJ
    
```

где AV — файл A3; n — младший адрес A3.

Еще одним приемом повышения информационной плотности может служить использование обратной магнитной поверхности ГМД (он хорошо известен многим программистам). Для этого необходимо с двух сторон конверта на месте отверстия под датчик начала дорожки прорезать овальное отверстие, похожее на прорезь для записи-считывания, так как при установке ГМД обратной стороной старое отверстие не совпадает с датчиком. Обратную сторону ГМД перед использованием необходимо отформатировать на приводе, входящем в состав терминала ЕС 8501M (терминал ЕС 8501M форматирует ГМД с одинарной плотностью).

Использование режима двойной плотности и второй магнитной поверхности позволяет увеличить информационную емкость НГМД в 4 раза.

Адрес для справок: г. Саратов-5, 410005, ул. Посадского, 235/243, кв. 77.

Статья поступила 16 февраля 1987 г.

РЕЖИМ ОЖИДАНИЯ В НАКОПИТЕЛЯХ НА ГИБКОМ МАГНИТНОМ ДИСКЕ

Широкое распространение персональных компьютеров и микропроцессорных (МП) систем немалым образом способствует развитию периферийного оборудования, в первую очередь без накопителей на гибких магнитных дисках (НГМД). Гибкие магнитные диски (ГМД), они же флопи-диски или дискеты, являются относительно новым, недорогим и очень удобным носителем информации. Наряду с функциями переноса информации дискеты могут служить и основным средством работы с ОС, хотя в большинстве профессиональных персональных компьютеров они используются только для связи пользователя с внешним миром, обмена информацией с другими пользователями. Для работы с файлами такие системы снабжаются накопителями большой емкости с жесткими дисками (например, винчестерского типа) или с гибкими дисками специальной технологии (например, типа «Бернулли»).

Использование же дискет в качестве основного средства работы с ОС и с файлами ведет к чрезмерному их износу.

Износ дискет вызывают: удары магнитной головки НГМД при опускании на вращающуюся поверхность ГМД в момент начала обмена информацией; стирание магнитного слоя гибкого диска из-за прижима магнитной головки к поверхности ГМД; задиры магнитного слоя (толщина которого не превышает 5 мкм), возникающие из-за нарушения поверхности магнитной головки (сколы, стирание, налипание частиц магнитного слоя), попадания посторонних частиц между поверхностью магнитного слоя и магнитной головкой, а также при износе или спекании специальной пылеулавливающей волокнистой подложки, которой выложена внутренняя поверхность конверта дискеты — именно о нее трется гибкий магнитный диск, вращаясь внутри конверта флопи-диска.

Износ волокнистого слоя находится в прямой зависимости от температуры, при которой работает дискета. Большинство дискет может работать в диапазоне температур +10°C...+52°C. Если внутри корпуса НГМД температура приближается к +50°C, трение ГМД о внутреннюю поверхность конверта дискеты может стать причиной превышения критического значения температуры — появятся нечитаемые зоны, которые идентифицируются ОС как FILE BAD (записанную в них информацию можно считать для пользователя утерянной).

Фирма BASF предлагает дискеты BASF FLEXY DISK SCIENCE (гибкие диски для научных исследований) для работ в компьютерах без кондиционирования воздуха. Их особенность — специальный материал конверта, позволяющий работать гибкому диску при температуре до +70°C. Время жизни информации гарантируется количеством проходов магнитной головки по каждой из дорожек для данного ГМД (70 млн). Дискеты BASF FLEXY DISK SCIENCE испытывались в целях абсолютной сохранности записанных данных с почти неограниченной продолжительностью использования. Однако стоимость этих дискет значительна и они недоступны подавляющему большинству пользователей.

Для изучения механизма износа, а также для оценки времени безотказной работы и проверки срока службы дискет в неэкстремальных условиях проводился эксперимент, в ходе которого на серийных НГМД «Электроника 6022» эксплуатировались мини-флопи-дискеты. Накопители входили в состав ПВМ ДВК-2М. Работа с дискетами длилась по 8...10 часов в сутки; один накопитель работал без кожуха и обдувался дополнительным вентилятором. Один комплекс ДВК-2М был поставлен на круглосуточную работу. На его дискетах каждые 15 минут протоколировалась информация о температуре и влажности окружающего воздуха. Помещение, где проходил эксперимент, не имело кондиционера, температура колебалась от +16°C ночью до +26°C днем. Относительная влажность воздуха менялась незначительно (60...70%). Результаты экспериментов приведены в таблице.

Анализ дискет, отказавших в процессе эксперимента, и их изучение под стереоскопическим микроскопом показали, что все нечитаемые

зоны на ГМД появились при повышенной температуре в области каталога и на дорожках, близких к центру дискеты.

В Советском Союзе широкое распространение получили ГМД двух размеров: дискеты Ø 200 мм (8 дюймов) используются в НГМД «Электроника 7012»; мини-флопи-диски, Ø 130 мм (5,25 дюйма) — в накопителях «Электроника 6022», «Электроника 6021», «Электроника 6011», «Электроника 6121» и др.

Первая цифра четырехзначного номера НГМД, выпускаемых электронной промышленностью, условно обозначает размер применяемого в накопителе диска, вторая — плотность информации при записи на ГМД: 0 соответствует одинарной плотности, 1 — двойной. Третья цифра говорит о количестве магнитных головок считывания-записи в каждом приводе накопителя (их может быть одна или две, причем две головки дают возможность доступа к обеим сторонам ГМД без манипуляций диском со стороны оператора). Последняя цифра — это количество приводов, объединенных в одном кожухе (иногда с общим блоком питания).

Таким образом, «Электроника 7012» — двоянный НГМД для дискет Ø 200 мм, каждый привод оснащен одной магнитной головкой записи-считывания, накопитель обеспечивает одинарную плотность, т. е. около 0,25М байт информации, записанной на 77 дорожках; «Электроника 6121» — одинарный привод с двумя головками считывания-записи, накопитель обеспечивает на дискете Ø 130 мм удвоенную плотность записи на 80 дорожках (900К байт информации); «Электроника 6022» — двоянный накопитель с одинарной плотностью, каждый из приводов оснащен двумя магнитными головками.

Диск должен вращаться не постоянно, а только во время обращения к нему, т. е. при записи или считывании.

Режим ожидания является важнейшим средством борьбы с преждевременным износом гибких магнитных дисков (ГМД), которому особенно подвержены мини-флопи-диски

Время безотказной работы дискет в серийном НГМД «Электроника 6022»

Тип	Число дискет, шт.	Длительность работы	
		число недель (8...10 часов в сутки)	число суток (круглосуточно)
Изот 5253-Е	3	5, 5, 8	4
ЕС 5288	3	7, 8, 8	5
TANDY	1	10	—
BASF	3	14	8
TA-96TPI	1	—	более 10

(Ø 130 мм). Сущность режима заключается в остановке маршевого двигателя накопителя на ГМД (НГМД) при отсутствии обращения к нему со стороны микроЭВМ.

Режим предусматривает: включение маршевого двигателя НГМД (конкретно выбранного привода, а не всех входящих в систему) по команде обращения;

выдержку 0,5 с перед опусканием головки считывания-записи на рабочую поверхность ГМД (для разгона маховика) до начала обмена информацией;

задержку выключения маршевого двигателя (2..5 с) после окончания обмена при повторном обращении к этому приводу;

останов маршевого двигателя после задержки и окончания обмена, либо при повторном обращении продолжение обмена без остановки двигателя.

В серийно выпускаемых НГМД «Электроника 6021» (6022, 6022А) применяются маршевые двигатели с непосредственным приводом, допускающие большое количество циклов включения-выключения, поэтому производительность микропроцессорной системы при работе НГМД в режиме ожидания снизится только на величину, обусловленную задержкой на раскрутку ГМД при первом обращении (0,3; 0,4 с). Последующие обращения (если они следуют без длительного перерыва) не вносят задержки, так как после окончания обращения ГМД продолжает вращаться 4, 5 с.

Все зарубежные МП системы, работающие с НГМД, используют этот режим — IN-USE [1, 2].

Серийный контроллер накопителей на гибких мини-флорн-дисках (КНГМД), которым до сих пор комплектуют ДВК-2М, ДВК-2МШ, ДВК-3, ДВК-3М2, к сожалению, не обеспечивает этого режима, хотя в последних моделях ДВК разработан интеллектуальный контроллер магнитного диска (КМД), поддерживающий режим ожидания [3]. Однако в МП системах, оснащенных КМД, невозможно раздельно управлять приводами НГМД «Электроника 6022», «Электроника 6022А» или «Электроника 6021». Это означает, что если в систему включен не один привод, а несколько (не считая двух дисководов Ø 200 мм с физическими адресами DX0; и DX1; в составе диалогового вычислительного комплекса без доработок контроллеров можно объединить еще 8 приводов, использующих диски Ø 130 мм: четыре управляются от КНГМД с физическими адресами MX0: ... MX3; и четыре — от КМД с физическими адресами MY0: ... MY3;), то при обращении к одному из четырех накопителей будут включены маршевые двигатели всех остальных, не считая излишнего расхода электроэнергии

(около 12 Вт на привод), напрасному износу из-за трения о внутреннюю поверхность дискеты подвергаются еще 3 гибких магнитных диска. В том случае, когда пользователь имеет в составе системы только КНГМД и работает не со специальным драйвером, а со стандартным, маршевые двигатели всех накопителей с именами МХ (N); запущенные в момент загрузки ОС, не останавливаются до момента выключения блока питания НГМД.

Остановить вращение ГМД в конверте дискеты можно вручную, подняв крышку накопителя. В подобном случае пользователь должен сам отслеживать сигнал «Обращение», открывая и закрывая крышки приводов накопителя в зависимости от загрузки красного светодиода на передней панели НГМД. При трансляции программ это еще допустимо, а как поступить при непрерывном круглосуточном сборе информации и записи ее на магнитный носитель?

К мерам, облегчающим температурный режим внутри накопителя, относится, в частности, вынос источника питания за пределы НГМД. В этом случае накопители потребляют энергию от центрального блока микроЭВМ.

Драйвер НГМД (или системная программа обслуживания накопителя) может реализовать режим ожидания, если монитор операционной системы разрешает тайм-аут с использованием системного таймера.

Устройство управления НГМД (расположено под кожухом накопителя) позволяет использовать несколько недействующих ИМС для доработки схемы устройства, необходимого для реализации режима. При этом в платы управления каждым приводом накопителя добавляются 2 резистора, 2 конденсатора и диод (рис. 1, 2).

Сигнал «Обращение», поданный на вход \bar{S} триггера управления двигателем, устанавливает выход триггера Q в состояние «Лог.1» и до тех пор,

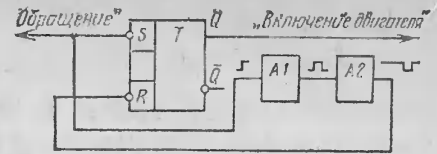


Рис. 1. Функциональная схема устройства

пока длится сигнал, удерживает его выход независимо от других сигналов, приходящих на триггер в процессе работы. Положительный перепад заднего фронта сигнала «Обращение» воспринимается введенным специально для этой цели одновибратором А1.

Одновибратор А1 вырабатывает импульс запуска для одновибратора А2, который обеспечивает задержку выключения триггера (и, следовательно, двигателя) на 4, 5 с. Это означает, что на выход одновибратора А2 можно подавать новый запускающий импульс, не дожидаясь отработки предыдущего. При этом выходной импульс с уровнем «Лог.1», не прерываясь, будет продолжен на величину задержки. Если в качестве одновибратора использовать ИМС К155АГ3, то величина задержки или длительность выходного импульса одновибратора А2, рассчитывается по формуле $T = KRC(1 + 0,7/R)$, ($K = 0,25$ для $R = 24..62$ кОм; $K = 0,47$ для $R > 64$ кОм; C в мкФ; T в мс) [4].

Задний фронт положительного выходного импульса одновибратора А2, выход которого соединен с входом R триггера управления маршевым двигателем, вызывает выключение триггера — выход Q триггера переходит из состояния «Лог.1» в состояние «Лог.0». Если на вход \bar{S} триггера к этому моменту опять не подан сигнал «Обращение», двигатель останавливается.

Элементы R59, R60, C36, C37, VD23 (рис. 3) напаяются в схеме навес-

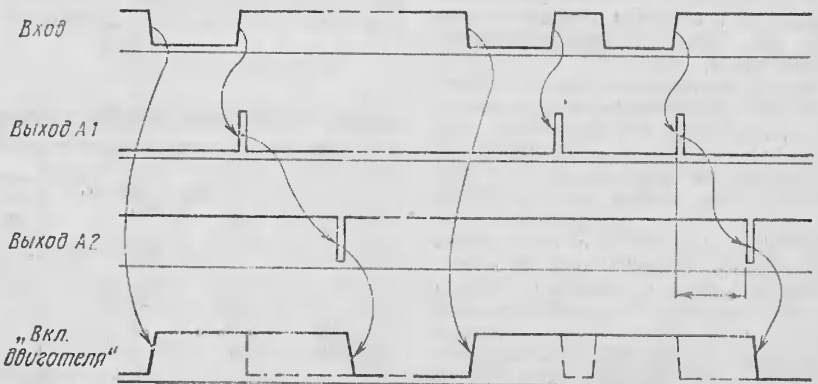


Рис. 2. Временные диаграммы

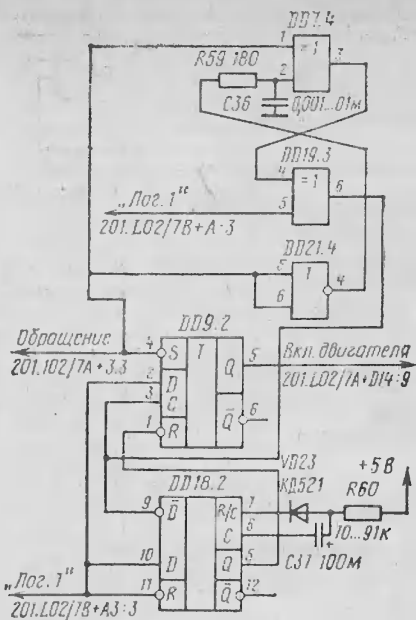


Рис. 3. Электрическая принципиальная схема устройства:

$C36=0,001 \dots 0,1$ мкФ, $C37=100,0$ мкФ, $R59=180$ Ом, $R60=91$ кОм

ным монтажом, переключатель Д1-Д2 выпадает. Все обозначения соответствуют обозначениям заводской схемы устройства управления М3.859.087.201 НГМД «Электроника 6022» [5].

Триггер DD9.2 выполняет функцию триггера управления двигателем; одновибратор DD18.2 — функцию одновибратора А2, а его времязадающие элементы C37, R60 и VD23 определяют длительность выходного импульса одновибратора или величину задержки выключения двигателя, одновибратор А1 реализован на элементах DD21.4, DD7.4, DD19.2, резисторе R59 и емкости C36.

Выдержку 0,5 с, необходимую для набора двигателем номинального числа оборотов перед опусканием магнитной головки на диск, реализует одновибратор с повторным запуском D18.1 схемы управления НГМД.

Схема не требует наладки и обеспечивает режим ожидания в полном объеме независимо от таймера, типа контроллера, версий системы и драйвера НГМД.

Срок службы дисков десятков накопителей, доработанных по приведенной схеме, увеличился в 7, 8 раз.

Иногда при разметке или инициализации нечитаемые зоны обнаруживаются на совершенно новых дисках без каких бы то ни было видимых причин. Использование односторонних дисков в накопителях с двумя головками записи-считывания (типичный случай — работа с дисками Изот 5253-Е, ЕС 5287, ЕС 5288 на НГМД «Электроника 6022» 6022А, 6021») приводит к возникновению нечитаемых зон. Лучший способ переноса и хранения дисков — в металлической коробке, которая экранирует ГМД от воздействия полей.

Большинство фирм-изготовителей ГМД маркируют диски следующими буквами:

SS/SD — SINGLE SIDE/SINGLE DENSITY (односторонняя дискета/одинарная плотность),
 SS/DD — SINGLE SIDE/DOUBLE DENSITY (односторонняя дискета/двойная плотность),
 DS/SD — DOUBLE SIDE/SINGLE DENSITY (двусторонняя дискета/одинарная плотность),
 DS/DD — DOUBLE SIDE/DOUBLE DENSITY (двусторонняя дискета/двойная плотность).

Встречается и другое обозначение диска (фирма BASF):

1X — односторонняя одинарная плотность,
 1D — односторонняя двойная плотность,

2X — двусторонняя одинарная плотность,

2D — двусторонняя двойная плотность,

2N — двусторонняя для приводов с одной головкой считывания-записи (с двумя маркерными отверстиями).

На мини-флопи-дискетах (Ø130 мм) иногда присутствуют цифры:

1/06 — односторонняя двойная плотность (96 дорожек на дюйм), способ записи MFM;

2/06 — двусторонняя двойная плотность (96 дорожек на дюйм), способ записи MFM.

Если этих цифр или обозначений нет, мини-флопи-дискета рассчитана на плотность 48 дорожек на дюйм (метод FM) и за попытки разметить методом MFM или MFM фирма-изготовитель ответственности не несет.

Телефон для справок: 483-72-17, Киев

ЛИТЕРАТУРА

1. Накопители на миниатюрном гибком магнитном диске ЕС 5088 и Изот-5050Е: Техническое описание К53.060.001 ТО. ГХО Изот. Болгария. — 1982. — С. 14.
2. Гибкий мини-диск К5600.2: Техническая документация. ГДР. Карл-Маркс-Штадт. 1986.
3. Бартевьев А. А., Беляев Е. Г. Программируемый контроллер для накопителя на гибких мини-дисках // Микропроцессорные средства и системы. — 1986. — № 4. — С. 46.
4. Данилов Р. В. и др. Применение интегральных микросхем в электронной вычислительной технике. 1986.
5. Накопитель автономный двоянный «Электроника 6022»: Техническое описание и инструкция по эксплуатации ДЕМ3.853.123.01 ТО.

Статья поступила 16 декабря 1986 г.

УДК 681.323

Я. Ф. Кисилевский, С. Г. Пикусов

ИСПОЛЬЗОВАНИЕ ДВОЙНОЙ ПЛОТНОСТИ ЗАПИСИ НА НГМД «ЭЛЕКТРОНИКА ГМД-7012» ПРИ РАБОТЕ С МИКРОЭВМ «ЭЛЕКТРОНИКА МС 1201»

Накопители на гибких магнитных дисках, входящие в комплект поставки ДВК, имеют емкость, недостаточную для работы с языками Фортран, Паскаль и т. д. Например, при емкости дискеты 250К байт (486 блоков после инициализации каталога посетителя) на ней не удастся разместить одновременно системные программы, занимающие приблизительно 110 блоков, транслятор с языка Фортран полной конфигурации (230) блоков, объектную библиотеку (210 блоков) и компоновщик

(45 блоков). Кроме того, на системном устройстве желательно иметь редактор текстов, отладчик, программы копирования файлов, распечатки каталогов и некоторые другие программы.

Выходом из этого положения явилось бы увеличение плотности записи на дисках. НГМД «Электроника ГМД-7012», входящие в состав диалоговых вычислительных комплексов ДВК-2, работают в двух режимах. Режим 1 — одинарная плотность записи (этот режим используется в ДВК). Режим 2 — одинарная и двойная плотность записи в зависимости от состояния восьмого разряда слова команды. Режим работы выбирается положением переключки Е1 на плате контроллера КЗ НГМД [1]. Интерфейс НГМД, входящий в состав микроЭВМ и выполненный на БИС К1801ВП1-033, не имеет восьмого разряда в регистре команд и состояний (РКС). Кроме того, в режиме 1 при передаче команды контроллер НГМД ожидает послыку из 8 бит, а в режиме 2 из 12 бит. Интерфейс же всегда обменивается с контроллером послысками из 8 бит и, таким образом, в режиме 2 контроллер не может правильно завершить процедуру приема команды.

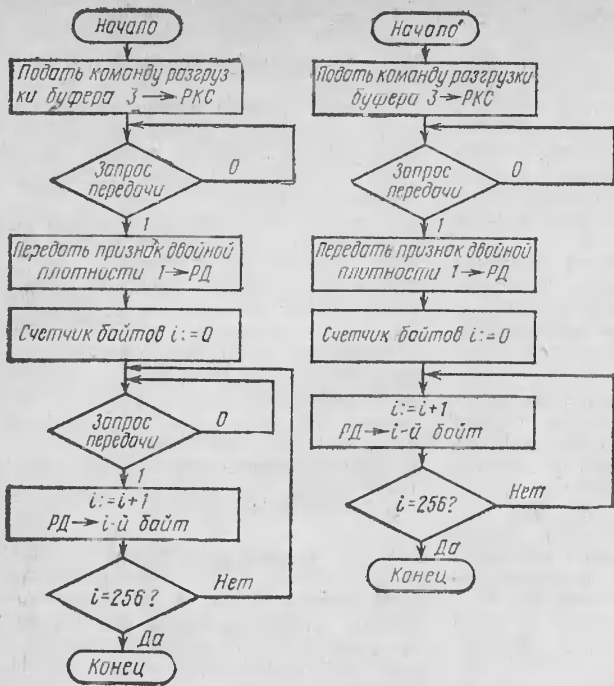


Рис. 1. Цикл обмена с ожиданием готовности

Рис. 2. Цикл обмена, реализованный в драйвере, работающем с двойной плотностью записи

Таким образом, аппаратная реализация НГМД позволяет использовать двойную плотность записи, но для этого необходимо изменить процедуру обмена. Прием заключается в том, что команда передается в два этапа: сначала пересылается восемь младших разрядов команды, затем старшие. Существенно, что первый байт команды пересылается по адресу РКС интерфейса НГМД, а второй — по адресу регистра данных (РД) НГМД. Таким образом, интерфейс передает в контроллер НГМД две посылки из 8 бит, а контроллер формирует из них необходимые ему 12 бит. Первый пересылаемый байт содержит собственно команду, а во втором байте значащим является только младший разряд, несущий информацию о плотности записи (0 — одна, 1 — двойная плотность).

Аргументы команды передаются обычным образом. Все команды без исключения должны пересылаться двумя байтами. Команды, для которых состояние восьмого разряда не существенно (например, чтение регистра ошибок), могут содержать во втором байте произвольную информацию, но передаваться он должен обязательно, иначе процедура приема команды не сможет завершиться или выполниться неправильно.

При разработке драйвера в ОС РТ-11 можно воспользоваться текстом системного драйвера DY.SYS, заменив в нем процедуру передачи команд на описанную выше. Для команд записи-чтения буфера сектора вместо команд передачи счетчика слов и адреса буфера обмена нужно построить свой цикл обмена, используя счетчик слов в качестве параметра цикла. При организации такого цикла необходимо учитывать, что счетчик слов обмена может содержать произвольное значение, не кратное 64 (для одинарной плотности) или 128 (для двойной), тогда как контроллеру должно быть передано или принято от него всегда 128/256 байт. В процедуре записи буфера сектора недостающие до кратного 128/256 числа байты должны быть переданы нулями, а в процедуре чтения необходимо обеспечить фиктивный ввод нужного числа байтов.

```

TITLE YVBOOT
ACSEMBLER MACRO-11 / RT-11
*****
* YVBOOT - АНАЛОГ АППАРАТНОГО ЗАГРУЗЧИКА. *
* ИСПОЛЬЗУЕТСЯ ДЛЯ ЗАГРУЗКИ ОПЕРАЦИОННОЙ *
* СИСТЕМЫ С ДИСКА С ДВОЙНОЙ ПЛОТНОСТЬЮ. *
*****

-FCALL -MODULE
MODULE YVBOOT, RELEASE=Y06, VERSION=06, COMMENT=(ГЛАВ.2. АППАРАТНЫЙ ЗАГРУЗЧИК)
.DSABL, OBL

УКД = 1 ; ИСПОЛЬЗУЕТСЯ ПРОЦЕССОР 1801ВН1
;
; АДРЕСА РЕГИСТРОВ
КВ.СБР = 177560 ; РКС КЛАВИАТУРЫ
ТТ.СБР = 177564 ; РКС ЭКРАНА
УУ.СБР = 177170 ; РКС ДИСКОВОДА

; РАЗРЫМ УПРАВЛЕНИЯ
С800 = 1 ; ПУСК
С801 = 28 ; ВЫБОР ПРИВОДА
С802 = 40 ; ЗАВЕРШЕНО
С811 = 208 ; ТРЕБОВАНИЕ ПЕРЕДАЧИ
С812 = 40000 ; СВРС

; КОМАНД
;
; РАЗРУЖКА БУФЕРА
С821 = 1*2 ; ЧТЕНИЕ СЕКТОРА
С822 = 3*2 ; СВРС

; МАКРОКОМАНДА ОЖИДАНИЯ ЗАВЕРШЕНИЯ ПЕРЕДАЧИ
LOOP1 MACRO YVREQ ?LOOP ;
TSIB 0R4 ; ПЕРЕДАЧА ЗАВЕРШЕНА ?
DPL LOOP ; НЕТ, ОЖИДАТЬ.
.ENDM

; МАКРОКОМАНДА ОЖИДАНИЯ ЗАВЕРШЕНИЯ ОПЕРАЦИИ
LDOPI MACRO YVDONE ?LOOP ;
BIT ?CSDONE, 1OR4 ; ОПЕРАЦИЯ ЗАВЕРШЕНА ?
BEQ LOOP ; НЕТ, ОЖИДАТЬ.
.ENDM

.ABECT
=10000
.ENABL

YVBOOT
MOV @YVBOOT, SP ; УКАЗАТЕЛЬ АДРЕСА ЗАГРУЗЧИКА
MTPB 0340 ; УСТАНОВИТЬ УКАЗАТЕЛЬ СТЕКА
MOV @MODULE, R1 ; ЗАПРЕТИТЬ ПЕРЕРЫВАНИЯ
MTPB @TT.CSR ; УКАЗАТЕЛЬ СОВМЕЩЕНИЯ -> R1
SPL 1* ; ОЖИДАТЬ ГОТОВНОСТЬ ЭКРАНА
MOV (R1)+, @TT.CSR+2 ; ВВЕСТИ СИМВОЛ НА ЭКРАН
SPL 1* ; ПОВТОРИТЬ, ЕСЛИ НЕ КОНЕЦ СООБЩЕНИЯ
MTPB @KB.CSR ; ОЖИДАТЬ ГОТОВНОСТЬ КЛАВИАТУРЫ
MOV R0, @TT.CSR+2 ;
SPL 2* ; ВВЕСТИ СИМВОЛ -> R0
MTPB @KB.CSR ; ВВЕСТИ СИМВОЛ НА ЭКРАН
MOV R0, @TT.CSR+2 ; R0 = НОМЕР ПРИВОДА
BIC @177776, R0 ; СОХРАНИТЬ НОМЕР ПРИВОДА
MOV R0, @UNIT ; ВМЕСТО СИМВОЛА ЧТЕНИЯ
ASL R0 ; ВМЕСТО СИМВОЛА ЧТЕНИЯ
MOV @YU.CSR, R4 ; АДРЕС РКС -> R4
MOV @YU.CSR+2, R5 ; АДРЕС БУФЕРА -> R5
MOV @C8RESET, 0R4 ; СВРС
YVDONE MOV 01, FWDENS ; УСТАНОВИТЬ ДВОЙНУЮ ПЛОТНОСТЬ
MOV 02, R0 ; ЧИСЛО СЕКТОРОВ ЧТЕНИЯ -> R0
MOV 0256, R3 ; ЧИСЛО БАЙТ ЧТЕНИЯ -> R3
TST FWDENS ; ОДИНАРНАЯ ПЛОТНОСТЬ ?
JNE 4* ; ДА, ПЕРЕЙТИ К 4*
ASR R3 ; НЕТ, ЧИСЛО СЕКТОРОВ = 4
CLR R2 ; ЧИСЛО БАЙТ = 312.
MOV 01, R1 ; АДРЕС ЗАГРУЗКИ -> R2
MOV (PC)+, 0R4 ; АДРЕС СЕКТОРА -> R1
REDCMD YVREQ ; ПЕРЕДАТЬ КОМАНДУ "ЧТЕНИЕ"
MOV (PC)+, 0R5 ;
YVDONE YVREQ ; ОЖИДАТЬ ЗАПРОС
MOV 1, FWDENS ; ПЕРЕДАТЬ ПЛОТНОСТЬ
YVREQ MOV R1, 0R5 ; ОЖИДАТЬ ЗАПРОС
YVREQ MOV 01, 0R5 ; ПЕРЕДАТЬ НОМЕР СЕКТОРА
YVDONE TST 0R4 ; ОЖИДАТЬ ЗАПРОС
DPL 7* ; ПЕРЕДАТЬ ЧИСЛО СЕКТОРОВ ?
BIT 020, 0R5 ; НЕТ, ПЕРЕЙТИ К 7*
JNE 4* ; СИМВОЛ ПЛОТНОСТИ ?
MOV @MERR, R1 ; ДА, ПЕРЕЙТИ К 4*
SPL 1* ; СООБЩЕНИЕ ОБ ОШИБКЕ -> R1
CLR FWDENS ; ПОВТОРИТЬ СНАЧАЛА
DR 3* ; ИЗМЕНИТЬ ПЛОТНОСТЬ
YVREQ MOV @C8BUF+C800, 0R4 ; ПОВТОРИТЬ ЧТЕНИЕ
MOV FWDENS, 0R5 ; ПЕРЕДАТЬ КОМАНДУ "РАЗГРУЗКА"
MOV R3, -(SP) ; ПЕРЕДАТЬ ПЛОТНОСТЬ
YVREQ MOV FWDENS, 0R5 ; СОХРАНИТЬ R3
IF DF W1 ; ОЖИДАТЬ ЗАПРОС
BOB R3, 7* ; ЧИТАТЬ R3 РАЗ (БЕЗ ПРОВЕРКИ
; ГОТОВНОСТИ)
; ЧИТАТЬ R3 РАЗ (С ПРОВЕРКОЙ
; ГОТОВНОСТИ)
IFDF BOB R3, 8* ; ГОТОВНОСТИ)
ADD 02, R1 ; УВЕЛИЧИТЬ НОМЕР СЕКТОРА
MOV (SP)+, R3 ; ВОССТАНОВИТЬ R3
BOB R0, 5* ; ЧИТАТЬ СЛЕДУЮЩИЙ СЕКТОР
MOV (PC)+, R3 ; ПЕРЕДАТЬ В R0 НОМЕР ПРИВОДА
SUNIT MOV 0, ;
JMP 000 ; ПЕРЕДАТЬ УПРАВЛЕНИЕ НАЧАЛЬНОМУ
; ЗАГРУЗЧИКУ
; КОД КОМАНД ЧТЕНИЯ С ПРИВОДА 0
; КОД КОМАНД ЧТЕНИЯ С ПРИВОДА 1
UNITREDCMD WORD C800+C8RD ;
WORD C800+C8RD+C8UNIT ;
.DSABL LSB

MERR .ABCI1 (15)(12)*YVBOOT-F-(16)*ОШИБКА ЧТЕНИЯ(17)
MQUEI .ABCI1 (15)(12)*YVBOOT-Q-(16)*НОМЕР ПРИВОДА ?-(17)(200)
.EVEN
.END YVBOOT

```

Рис. 3. Программа-аналог аппаратного загрузчика

Если строить цикл обмена обычным образом с ожиданием завершения передачи байта (рис. 1), то оказывается, что для микроЭВМ «Электроника МС 1201» с процессором К1801ВМ1 при работе с двойной плотностью время выполнения цикла несколько превышает время прохождения физического сектора диска вдоль головки записи-чтения. Это приводит к тому, что за один оборот диска можно считать или записать данные только на один сектор, что существенно замедляет процедуру обмена. Время обмена одним байтом между контроллером НГМД и интерфейсом не превышает 8 мкс, что приблизительно равно времени выполнения одной команды пересылки процессора К1801ВМ1. Это позволяет отказаться от проверки завершения обмена байтом и тем самым сократить число команд в цикле и время его выполнения (рис. 2). Если ЭВМ оснащена процессором К1801ВМ2 или более быстродействующим, то проверку завершения в теле цикла следует оставить. Выбор варианта цикла в зависимости от типа процессора можно включить в программу директивы условной трансляции.

Дискета для работы в режиме с двойной плотностью должна быть подготовлена соответствующим образом: чистую отформированную дискету вставляют в дисковод и в пультовом режиме передают следующую последовательность команд:

Привод 0	Привод 1
11 → РКС	31 → РКС
1 → РД	1 → РД
111 → РД	111 → РД

При этом во все зоны данных на дискете будут записаны нули с признаком двойной плотности. Для загрузки операционной системы с такого диска штатный аппаратный загрузчик не пригоден, поэтому используется аналог аппаратного загрузчика, текст которого приведен на рис. 3.

В заключение следует отметить, что при работе с дискетами в режиме двойной плотности записи наряду с увеличением емкости вдвое повышается скорость обмена с накопителем, так как за один оборот дискеты с нее считывается или записывается вдвое больше информации. Описанный алгоритм обмена был проверен в работе с шестью НГМД разных годов выпуска и хорошо себя зарекомендовал. Сбоев в работе не обнаружено.

Телефон для справок: 261-51-02, Киев

УДК 681.326.3

И. Л. Мышкин, А. С. Щербаков

ИНТЕРФЕЙС НАКОПИТЕЛЯ НА ГИБКИХ МАГНИТНЫХ ДИСКАХ «ЭЛЕКТРОНИКА ГМД-7012»

Аппаратно-программный интерфейс предназначен для сопряжения микроЭВМ на базе микропроцессора КР5801К80А с НГМД «Электроника ГМД-7012». Аппаратная часть содержит одну БИС программируемого параллельного интерфейса (ППИ) КР5801К55 Д1 и восемь микросхем средней степени интеграции (рис. 1).

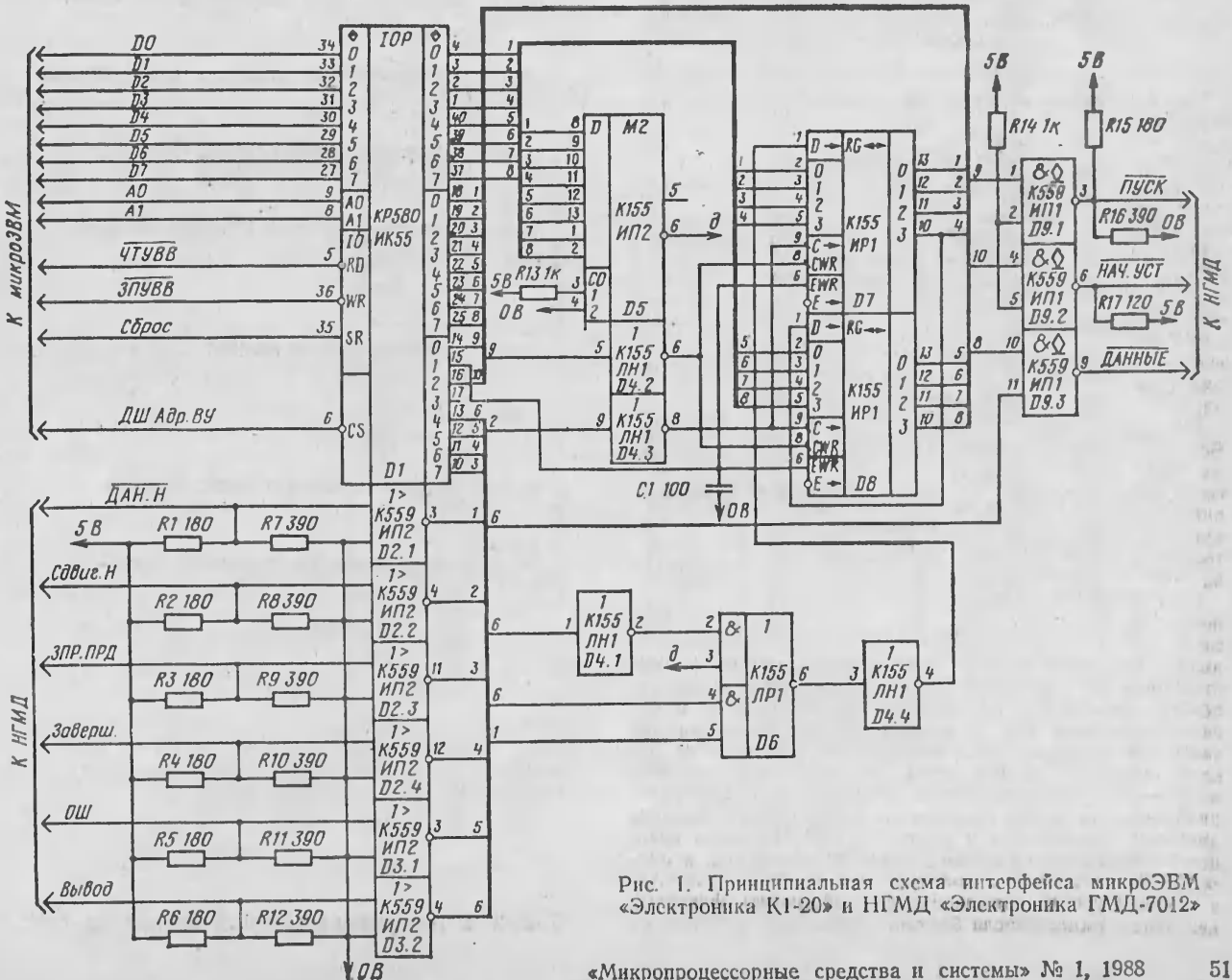


Рис. 1. Принципиальная схема интерфейса микроЭВМ «Электроника К1-20» и НГМД «Электроника ГМД-7012»

```

.TITLE PL
.MCALL .GELDF, .DRBEG, .DRAST, .DRFIN, .DREND
.MCALL .FORK, .INEN
.IIF NDF MMGT, MMGT=0 ; ПАРАМЕТРЫ ГЕНЕРАЦИИ ДОЛЖИ
.IIF NDF ERLG, ERLG=0 ; СООТВЕТСТВОВАТЬ СЛОВУ
.IIF NDF TIMKT, TIMKT=0 ; ГЕНЕРАЦИИ СИСТЕМЫ SYSGEN
.GELDF
WCNT=Q.WCNT-Q.BLKN
BUFF=Q.BUFF-Q.BLKN
LINK=Q.LINK-Q.BLKN
PLDSIZ=0 ; РАЗМЕР УСТРОЙСТВА
PLSTS=357 ; СЛОВО СТАТУСА УСТРОЙСТВА
PLICS=176560 ; АДРЕСА РЕГИСТРОВ
PLIBR=PLICS+2
FLOCS=PLICS+4
PLOBR=PLICS+6
PLIVEC=160 ; ВЕКТОРЫ ПЕРЫВАНИИ
PLOVEC=PLIVEC+4
MDERR=1
; ИНИЦИАЛИЗАЦИЯ ВВОДА-ВЫВОДА
.DRBEG
JSR PC,ADRSET
TST @#PLIBR
ROWR: TST WCNT(R4) ; ВВОД ИЛИ ВЫВОД?
BPL INPT
ASL WCNT(R4) ; ВЫВОД
BCC ERR
BEQ DONE
CLR READ
ORET: BIS #100,@#FLOCS
IRET: BIS #100,@#PLICS
RTS PC
INPT: ASL WCNT(R4) ; ВВОД
BEQ DONE
MOV #1,READ
TST @#PLIBR
TST RECD ; ПУСТ ЛИ БУФЕР?
BNE INPFIN ; ЕСЛИ ЖДАТЬ, ТО
TST WAIT ; РАЗРЕШИТЬ ПЕРЫВАНИИ
BNE IRET
BR EOF
INPFIN: CLR INPCNT ; ПЕРЕДАЧА ВВЕДЕННОЙ
JSR PC,ADRSET ; ИНФОРМАЦИИ МОНИТОРУ
PASS: MOVB @BUFF,INBUF
JSR PC,NEXT
INC BUFP
INC INPCNT
CMP INPCNT,RECD
BLT PASS
CLR INPCNT
CLR RECD
CLR WAIT
;
JSR PC,EOLN
JSR PC,EOLN
EOF: CLR INBUF
JSR PC,NEXT
TST WCNT(R4)
BGT EOF
BNE PDONE
EOLN: MOV #15,INBUF
JSR PC,NEXT
MOV #12,INBUF
JSR PC,NEXT
RTS PC
NEXT: .IF EQ MMGT
.IFT MOVB INBUF,@BUFF(R4)
INC BUFP(R4)
.IFF MOV INBUF,-(SP)
JSR PC,@XTBYT
.ENDC
DEC WCNT(R4)
RTS PC
ERR: BR ERROR
;DRAST FL,4,PDONE ; ПЕРЫВАНИЕ ПО ПЕРЕДАЧЕ
MOV PLQCE,R4
TST @#FLOCS
BMI ORET
TSTB @#FLOCS
BPL ORET
CLR @#FLOCS
CLR PLFBLK

```

```

WRB0: LDA IOF ; УСТАНОВКА АДРЕСА ДАННЫХ В ОЗУ
LXI H,IOD
OUT PADISK ; ВЫВОД КОМАНДЫ
CALL WREG ; ПОДАЧА СИГНАЛА 'ПУСК'
WRB2: CALL WTZP ; ОЖИДАНИЕ СИГНАЛА 'ЗАПРОС ПЕРЕДАЧИ'
MOV A,M ; ЧТЕНИЕ БАЙТА ИЗ ПАМЯТИ
OUT PADISK ; ВЫВОД В РЕГИСТР НГМД
CALL WREG ; ПОДАЧА ИМПУЛЬСА 'ПУСК'
WRB3: CALL INZP ; ЕСЛИ СНЯТ ЗАПРОС ПЕРЕДАЧИ
JNZ NKTBYT ; ПЕРЕЙТИ К СЛЕДУЮЩЕМУ БАЙТУ
CALL INZAV ;
JZ WRB3 ; ЕСЛИ НЕТ ЗАВЕРШЕНИЯ ПОВТОРИТЬ ЦИКЛ
CALL INERR ; АНАЛИЗ ОШИБКИ
RZ ; ВОЗВРАТ БЕЗ ОШИБКИ
DCR C
JNZ WRB0 ; ПОВТОРЕНИЕ ОПЕРАЦИИ ПРИ ОШИБКЕ
JMP ERDISK ; ВЫХОД ПО ОШИБКЕ
NKTBYT: INX H ; ПЕРЕХОД К СЛЕДУЮЩЕМУ АДРЕСУ ОЗУ
JMP WRB2 ; ВОЗВРАТ В ЦИКЛ ВЫВОДА
;
WREG: ; ПОДПРОГРАММА ЗАПИСИ БАЙТА В РЕГИСТР НГМД
; И УСТАНОВКИ СИГНАЛА 'ПУСК'
MVI A,PARAL
OUT PRDISK ; УСТАНОВКА ПАРАЛЛЕЛЬНОГО ЗАНЕСЕНИЯ
ANI MPUSK ; ТРАНСЛЯЦИЯ КОДА КОМАНДЫ 'ПУСК'
OUT PRDISK ; УСТАНОВКА СИГНАЛА 'ПУСК'
XRA A
OUT PRDISK ; СНЯТИЕ СИГНАЛА 'ПУСК'
RET
INZP: ; ПОДПРОГРАММА ВВОДА СОСТОЯНИЯ СИГНАЛА 'ЗПР. ПЕРЕДАЧИ'
IN PCDISK
ANI MZP
RET
INZAV: ; ПОДПРОГРАММА ВВОДА СОСТОЯНИЯ СИГНАЛА 'ЗАВЕРШЕНО'
IN PCDISK
ANI MZAV
RET
INUVU: ; ПОДПРОГРАММА ВВОДА СОСТОЯНИЯ СИГНАЛА 'ВЫВОД'
IN PCDISK
ANI MUVU
RET
INERR: ; ПОДПРОГРАММА ВВОДА СОСТОЯНИЯ СИГНАЛА 'ОШИБКА'
IN PCDISK
ANI MERR
RET
WTZP: ; ПОДПРОГРАММА ОЖИДАНИЯ СИГНАЛА 'ЗАПРОС ПЕРЕДАЧИ'
CALL INZP
JZ WTZP
RET
WTZAV: ; ПОДПРОГРАММА ОЖИДАНИЯ СИГНАЛА 'ЗАВЕРШЕНО'
CALL INZAV
JZ WTZAV
RET
NUST: ; ПОДПРОГРАММА НАЧАЛЬНОЙ УСТАНОВКИ ДИСКОВОДА
MVI A,USTDISK
OUT PRDISK
DCR A
OUT PRDISK
CALL WTZAV
RET
FROS: ; ПОДПРОГРАММА ЧТЕНИЯ РОС НГМД
MVI A,RROSE
OUT PRDISK
CALL WREG
ROS2: CALL INZAV
JZ ROS2
IN PBDISK
RET
ERDISK: ; ПОДПРОГРАММА ВЫХОДА ПО ОШИБКЕ ДИСКОВОДА
MVI A,1
RET
SETFUNK: ; УСТАНОВКА КОДА КОМАНДЫ ДЛЯ ПОСЛЕДУЮЩЕГО ВВ-ВЫВ
; КОМАНДА В РЕГ. С)
LXI H,IOF
MOV A,M
ANI 11111000B
ORA C
MOV M,A
RET
;
;
;
; ОБЛАСТЬ ДАННЫХ (ДОЛЖНА РАСПОЛАГАТЬСЯ В ОЗУ)
IOF: DB READ ; ФУНКЦИЯ ВВ-ВЫВ
IGN: DB 1 ; К-ВО ЧИТАЕМЫХ СЕКТОРОВ
IOT: DB 1 ; НОМЕР ДОРЖИКИ
IOS: DB 1 ; НОМЕР СЕКТОРА
IOD: DW BUFF ; АДРЕС ОЗУ ДЛЯ ВВ-ВЫВ
;
;
END

```

С выхода порта А ППИ данные поступают на формирователь бита четности D5 и параллельные входы регистра сдвига D7, D8, который преобразует их в последовательный код. С выхода регистра через магистральный усилитель D9.3 информация поступает на линию «данные». С нее через магистральный приемник D2.1 — на последовательный вход регистра. Стробирование сдвига осуществляется по линии «сдвиг». Выходы регистра подключены к порту В ППИ D1, через который обеспечивается прием данных от накопителя в микроЭВМ.

Порт С ППИ служит для формирования служебных сигналов микроЭВМ и приема служебных сигналов от

Рис. 2. Программа-драйвер накопителя на ГМД

накопителя. Разряд С0 вырабатывает сигнал «пуск», по которому данные с параллельных входов записываются в регистр D7, D8 и дается разрешение накопителю на считывание. Разряд С1 разрешает регистру параллельное записывание. Этот сигнал снимается синхронно с подачей сигнала «пуск». Разряд С2 формирует сигнал «начальная установка».

Разряды С4...С7 принимают от накопителя сигналы «вывод», «ошибка», «завершено» и «запрос передачи» соответственно. Сигнал «вывод» переключает коммутатор D6, который в режиме вывода подает на вход сдвигового регистра данные от накопителя (в режиме ввода подается бит четности) и отключает магистральный передатчик D9.3 от линии «данные».

Драйвер накопителя, написанный на языке ассемблера (рис. 2), состоит из двух основных подпрограмм: READ — чтения сектора с ГМД в ОЗУ микроЭВМ и WRITE — записи данных из ОЗУ на диск.

Для обращения к драйверу из программы пользователя необходимо заполнить таблицу параметров ввода-вывода IOPB. Пользователь должен заполнить ячейку номера дорожки IOT и номера первого сектора IOS, а также два байта IOD адресом начала буфера в ОЗУ микроЭВМ.

Если операция обращения к накопителю прошла нормально, то драйвер возвращает в аккумулятор МП код 00, а если произошла ошибка, то повторяет операцию до 10 раз. В случае неудачного исхода возвращает в аккумулятор код 01 с установленным признаком ненулевого аккумулятора.

Интерфейс использовался авторами на микроЭВМ «Электроника К1-10» при запуске операционной системы ОС 1800.

В случае подключения интерфейса к микроЭВМ «Электроника К1-20», микросхему ППИ D1 можно

исключить и пользоваться параллельными каналами ввода-вывода. В соответствии с этим необходимо в драйвере указать реальные адреса портов ППИ или каналов ввода-вывода.

Описанная схема аппаратно-программного интерфейса НГМД «Электроника ГМД-7012» позволяет относительно легко реализовать дисковую операционную систему практически на любой микроЭВМ с процессором КР580ИК80А.

Сообщение поступило 26 мая 1986 г.

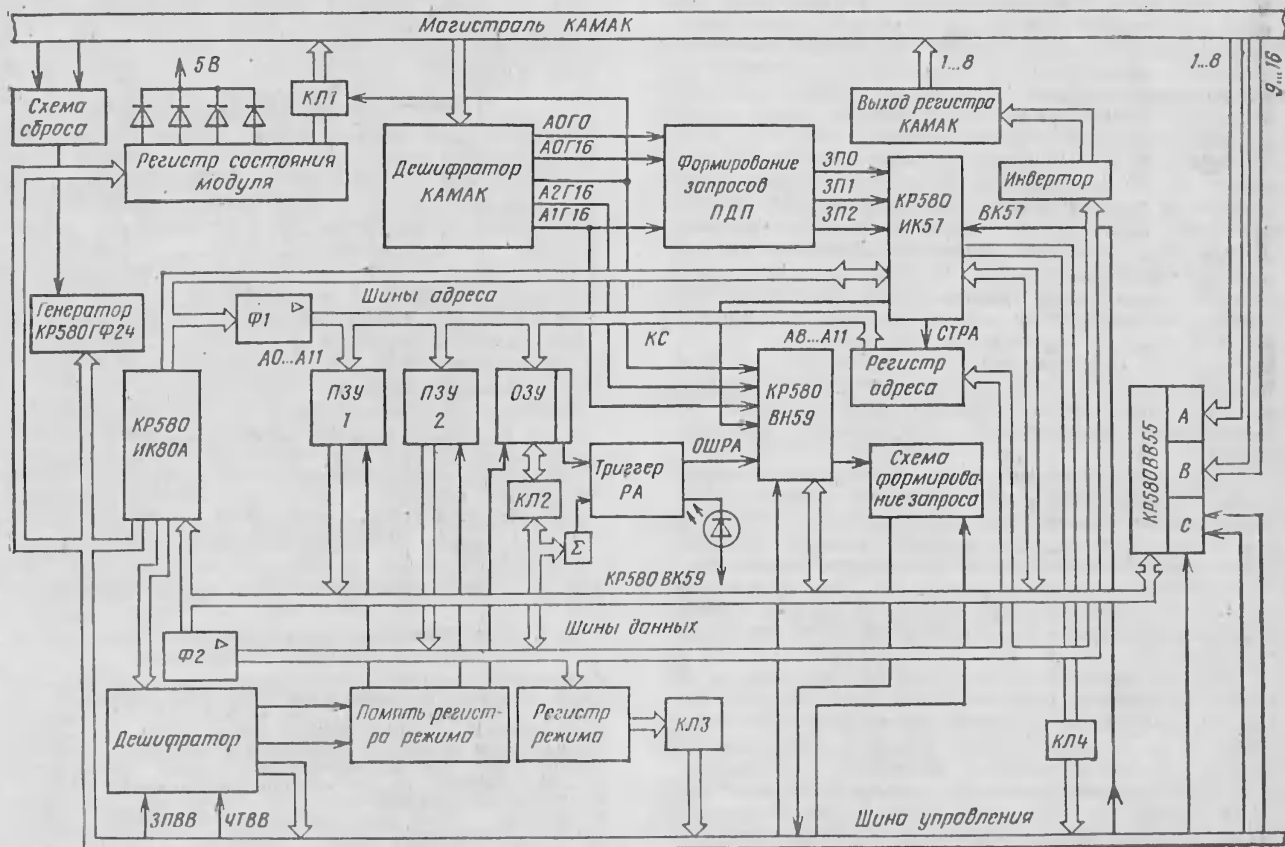
УДК 681.323

О. В. Кулик, М. А. Андронов

МОДУЛЬ СОПРЯЖЕНИЯ БИС СЕРИИ КР580 С МАГИСТРАЛЬЮ КРЕЙТА КАМАК

Интерфейсный модуль INTCAM-1 (см. рисунок) дает возможность организовать одновременную работу стандартных модулей КАМАК и периферийных БИС серии КР580 в одном крейте. Встроенный микропроцессор К580МК80А и ОЗУ К537РУ3 делают INTCAM-1 интеллектуальным модулем, который может использоваться для решения задач предварительной обработки экспериментальных данных. В отличие от модуля U-80* в модуле INTCAM-1 большинство узлов собрано на БИС серии КР580, что позволило расширить его функцио-

* Заушницин В. Л., Кулик О. В., Репин В. М. Унифицированный микропроцессорный модуль U-80 в стандарте КАМАК на базе КР580ИК80А. — ПТЭ. — 1984. — № 5. — С. 65—69.



Структурная схема интерфейсного модуля

нальные возможности, уменьшить потребляемую мощность и в итоге увеличить надежность.

В модуле INTSAM-1 дешифрируется всего пять команд КАМАК. Команды обмена данными реализованы аппаратно на основе контроллера ПДП, а команды управления и состояния модуля — программно по прерыванию. Такая структура позволяет при минимальных аппаратных затратах увеличить гибкость использования модуля без снижения скоростных характеристик. Модуль имеет ПЗУ объемом 2К байта и ОЗУ 4К байта с контролем четности.

УДК 681.325.621:397.61

А. Я. Суранов, П. И. Госьков, А. Г. Якунин

МИКРОПРОЦЕССОРНАЯ СИСТЕМА ИЗМЕРЕНИЯ ПАРАМЕТРОВ БИНАРНЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ МНОГОЭЛЕМЕНТНОГО ФОТОПРИЕМНИКА

Одно из перспективных направлений автоматизации процессов измерения и контроля геометрических параметров объектов связано с использованием оптоэлектронных теневого методов и соответствующих средств, в частности многоэлементных фотоприемников (МЭФ). На поверхности МЭФ формируется двухуровневое световое поле, геометрические параметры которого однозначно связаны с аналогичными параметрами объекта. Специфика светового поля предполагает выделение на выходе фотоприемника бинарного сигнала, который подвергается дальнейшей обработке в соответствии с назначением измерительной системы.

Помимо функций измерения такая система должна решать задачи адаптации к условиям измерения, взаимодействия с оператором и управляющей ЭВМ. Сложность этих задач в сочетании с требованием гибкости системы определяет необходимость применения в ней микропроцессорных средств. Система в этом случае становится достаточно универсальной, и для ее настройки на конкретную задачу достаточно изменить входную оптику и программное обеспечение.

Очевидно, максимальная универсальность системы достигается, когда для программной обработки используются данные, представляющие собой исходный бинарный или оцифрованный полутоновый видеосигнал [1].

Однако вычислительных ресурсов микроконтроллеров на базе микропроцессорных комплектов широкого применения часто недостаточно для обработки таких данных в реальном масштабе времени, поэтому особенно важно выбрать оптимальное соотношение между программными и аппаратными затратами, рационально организовать вычислительный процесс и процесс обмена информацией между микроконтроллером и внешними устройствами.

Предварительный отбор и обработку информации из исходного сигнала фотоприемника целесообразно выполнять аппаратными средствами по жесткому алгоритму, а на микроконтроллер возложить перенастраиваемые функции управления, адаптации и окончательной обработки малоразмерных массивов данных [2]. В частности, для бинарного изображения в качестве таких данных удобно использовать ширину, координаты и число проецируемых на МЭФ световых зон, определяемых путем подсчета элементов фотоприемника на

К модулю разработана приставка, предназначенная для автоматизации спектрометрических исследований. Приставка содержит три 10-разрядных ЦАП (время преобразования 5 мкс.), 10-разрядный АЦП (время преобразования 2 мкс.), три 16-разрядных счетчика, кварцевый генератор.

По вопросам получения более подробной информации просим обращаться по адресу: Москва, Ленинские горы, МГУ, физический ф-т, кафедра математики. Тел. 939-13-51.

Сообщение поступило 14 октября 1986 г.

ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ И УСТРОЙСТВА

соответствующих участках видеосигнала с учетом числа таких участков. Для согласования скорости формирования данных со скоростью их ввода в микропроцессор используется буферное запоминающее устройство (БЗУ), организованное в виде стека с возможностью программного доступа к регистру — указателю стека. На практике общее число световых зон обычно невелико, обмен данными с БЗУ можно выполнять программным путем, что существенно упрощает аппаратные затраты за счет исключения системы прерываний и системы прямого доступа к памяти.

Разработанная микропроцессорная система измерения параметров бинарных изображений на основе линейного ПЗС-фотоприемника К1200ЦЛ1 и микропроцессорного комплекта серии К580 (рис. 1) имеет блочно-мо-

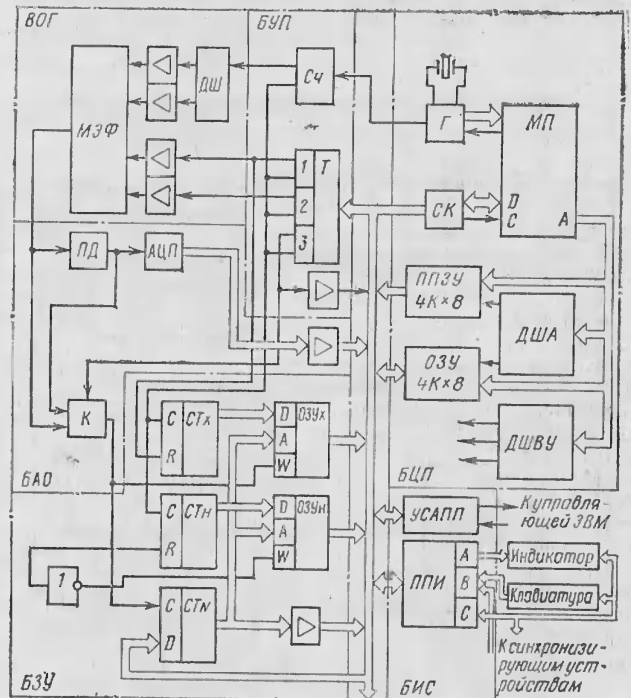


Рис. 1. Функциональная схема системы измерения параметров бинарных изображений.

МЭФ — многоэлементный фотоприемник (К1200ЦЛ1); ПД — пиковый детектор; К — компаратор; СТх, СТн — счетчики координаты и ширины световых зон (2×К155ИЕ5); Сч — счетчик числа световых зон (2×К155ИЕ7); ОЗУх, ОЗУн — оперативные запоминающие устройства координаты и ширины световых зон (2×К155РУ2); Т — программируемый таймер (КР530ВИ53); Сч, ДШ — счетчики и дешифратор фазных импульсов МЭФ; Г — генератор (КР580ГФ24); СК — системный контроллер (КР580ВК28); МП — микропроцессор (КР580ИК80); ДША, ДШВУ — дешифраторы адреса памяти и внешних устройств (2×К155РТ4); УСАП — универсальный системный адаптер последовательной передачи (КР530ВВ51); ППИ — программируемый интерфейс (КР580ВВ55).

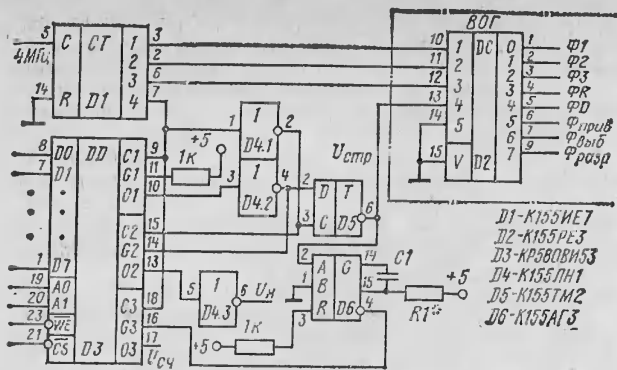


Рис. 2. Схема электрическая принципиальная блока управления фотоприемником:

$\Phi_1 - \Phi_3$ — фазные импульсы; Φ_R — импульс сброса выходного устройства; Φ_D — импульс управления входным устройством; $\Phi_{прив}$, $\Phi_{выб}$ — импульсы привязки и выборки видеосигнала; $\Phi_{разр}$ — импульс разрешения переноса заряда из фотосекции вдвигающий регистр

дальнюю структуру, состоит из модуля выносной оптической головки (ВОГ) с ПЗС-фотоприемником, схемой опроса и предусилителем и модуля обработки сигнала, конструктивно совмещенного с блоком питания. В состав модуля обработки входят блок центрального процессора (БЦП), выполненный на отдельной плате, блок интерфейсных схем (БИС), блок управления приемником (БУП), блок аналогового обработки (БАО) и БЗУ. Объем ОЗУ и ПЗУ по 4К байта.

Блок интерфейсных схем через универсальный связной адаптер последовательной передачи (УСАПП) обеспечивает обмен служебной информацией и результатами измерения с управляющей ЭВМ, а через программируемый параллельный интерфейс — с терминалом оператора и внешними устройствами.

Блок управления приемником (рис. 2) служит для формирования набора синхросигналов, задающих цикл работы устройства. Основным элементом БУП является программируемый таймер D3. Один из счетчиков работает в режиме деления частоты, вырабатывая строчные импульсы $U_{стр}$, задающие период считывания изображения $T_{стр}$ (рис. 3), два других — в режиме программируемого одновибратора. Один из них формирует импульс U_a длительностью T_a , определяющий момент начала опроса приемника и время экспозиции (накопления) $T_a = T_{стр} - T_z$, а другой — импульс $U_{сч}$ длительностью $T_{сч}$, равной времени считывания сигнала из МЭФ. Последний импульс используется для стробирования входящего в БАО компаратора К, переключения БЗУ из режима чтения в режим записи, а также выполняет роль сигнала готовности БЗУ к обмену данными с процессором. Фазные импульсы управления сканированием изображения МЭФ вырабатываются элементами D1 и D2 в соответствии с рекомендуемой в работе [3] временной диаграммой. С целью сокращения числа связей и повышения помехоустойчивости ПЗУ D2 размещено в ВОГ. Содержимое ПЗУ приведено ниже.

Прошивка ПЗУ D2

Адрес	Код
0	221
1	253
2	212
3	216
4	316
5	216
6	214
7	215
10	1
.	.
.	.
17	1

Блок управления обеспечивает опрос 1024 элементов изображения за время 2048 мкс, позволяет регулировать время накопления от 2 мкс до 129 мс и период опроса от 2 до 129 мс. При необходимости указанные временные соотношения могут быть легко смещены в ту или иную сторону путем выбора отличной от 4 МГц частоты синхронизации и введения между элементами D1 и D3 дополнительного делителя частоты. В любом случае время опроса элемента фотоприемника должно быть не менее 1 мкс, а время накопления — не более 0,2..0,5 с.

Система адаптации, предназначенная для формирования бинарного видеосигнала кроме микропроцессора и программируемого одновибратора включает пиковый детектор и малоразрядный АЦП, квантующий сигнал с детектора в окрестности его номинального уровня. Фиксируемое пиковым детектором максимальное значение сигнала по окончании считывания изображения текущей строки преобразуется в цифровую форму и через буферный регистр вводится в центральный процессор, осуществляющий расчет и установку оптимального времени накопления для очередного кадра. Одновременно максимальное значение сигнала в предыдущей строке задает величину порога U_p , по которому производится компарирование считываемого сигнала изображения U_c текущей строки (см. рис. 3). Таким образом, адаптация способом варьирования порога срабатывания компаратора обеспечивает высокие динамические характеристики системы при резких случайных изменениях уровня сигнала.

Адаптация по времени накопления обеспечивает в установленном режиме при малых флуктуациях освещенности высокую точность отслеживания, оптимизирует отношение сигнал/шум, позволяет оценивать абсолютное значение оптического сигнала, которое также может служить информационным параметром.

Бинарный сигнал U_b с выхода БАО используется для управления элементами БЗУ. В состав последнего помимо запоминающих устройств, рассчитанных на запись информации из 32 зон, входят счетчики координат ширины и числа объектов. Счетчик координаты СТх запускается в начале каждого кадра. По фронту бинарного сигнала производится запись координаты передней границы световой зоны в ОЗУх (рис. 3) и запуск счетчика ширины СТн. По срезу сигнала в ОЗУи записывается ширина световой зоны и одновременно инкрементируется содержимое счетчика СТн числа объектов. Частота заполнения счетчиков выбрана так, чтобы для любых видов сигналов не происходило переполнения 8-разрядных ОЗУн и ОЗУх. При этом абсо-

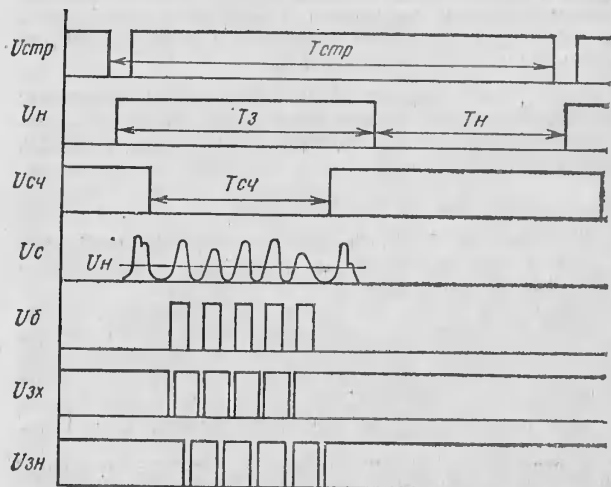


Рис. 3. Временные диаграммы работы элементов микропроцессорной системы

любая погрешность измерения не превышает размеров четырех элементов МЭФ, а приведенная — 0,5 %. Поскольку БАО обеспечивает фиксацию границ световых зон с точностью до одного элемента, погрешность измерения может быть легко уменьшена по крайней мере в 4 раза путем изменения частоты заполнения счетчиков СТх и СТн, а при применении в БАО более сложных алгоритмов обработки сигнала — и в большее число раз. При этом суммарная погрешность будет определяться уже не столько фотоприемником, сколько оптической схемой. Следует также иметь в виду, что снижение погрешности повлечет за собой реорганизацию структуры ОЗУ, поскольку информация о координате или ширине световой зоны в общем случае может занимать более одного байта.

Программное обеспечение системы включает основную подпрограмму считывания и обработки изображения и вспомогательные подпрограммы: системные драйверы и подпрограммы тестирования работоспособности устройства. Подпрограммы выполняются под управлением монитора, опрашивающего регистры состояния УСАПП и ППИ. По запросу от центральной ЭВМ или с клавиатуры оператора происходит переход к выполнению соответствующей подпрограммы. При работе с оптическим сигналом элементарное звено таких подпрограмм обычно состоит из циклов ожидания завершения загрузки БЗУ, выполнения процедуры адаптации, считывания информации из БЗУ и ее обработки. Все программное обеспечение, рассчитанное на решение конкретной задачи, записано в две микросхемы УФ РПЗУ К573РФ2, а ОЗУ служит лишь для хранения модифицируемых данных в процессе работы программ. Это позволяет при наличии терминала оператора применять устройство не только в качестве первичного измерительного преобразователя, но и как автономный прибор.

Подготовка и отладка программного обеспечения выполнялась в несколько этапов [4]. На первом этапе с помощью эмулятора БЦП на базе микроЭВМ «Электроника 60» отлаживался алгоритм обработки сигнала с использованием диалоговой системы программирования QUASIC. Затем для отлаженного алгоритма составлялся текст исходной программы на языке ассемблера, из которого с помощью кросс-системы [5] создавался загрузочный модуль. Отладка последнего выполнялась с помощью внутрисхемного эмулятора, аналогичного описанному в работе [6].

Система использовалась для решения задач считывания кодовых меток, контроля целостности нитей в текстильном производстве и контроля параметров зерна. Среднее время периода опроса МЭФ в перечисленных задачах составляло 10—20 мс, приведенная погрешность определения координат и размеров контролируемых объектов — менее 1 %.

Адрес для запроса дополнительной информации: 656099, Барнаул, проспект Ленина, 46. Алтайский политехнический институт, кафедра общей электротехники. Тел. 5-54-11.

ЛИТЕРАТУРА

1. Головцов А. Л. Ввод телевизионных изображений в ЭВМ-4 // Приборы и системы управления, — 1985, — № 5, — С. 10—12.
2. Госьков П. И., Якунин А. Г. Особенности разработки и средства программно-аппаратного обеспечения ОЭП на базе систем МЭФ МПК. // Оптические сканирующие устройства и приборы на их основе: Тезисы докл. III Всесоюз. совещ. — Барнаул: Алтайский политехнический институт, 1986, ч. 1, с. 158—167.
3. Кашлаков И. Д., Клепов В. Г., Костюков Б. В. Линейная фоточувствительная схема с зарядовой связью К1200 ЦП1 // Электронная техника, — 1982, — Вып. 7, — С. 7—9.

4. Суранов А. Я., Холупко Ю. Б., Якунин А. Г. Система разработки программного обеспечения микропроцессорных контроллеров оптико-электронных устройств на базе вычислительного комплекса «Электроника 100/15» — «Электроника 60». // Оптические сканирующие устройства и измерительные приборы на их основе: Тезисы докл. III Всесоюз. совещ. — Барнаул: Алтайский политехнический институт, 1986, ч. 1, с. 177—178.
5. Лукьянов Д. А. Как написать кросс-транслятор с языка ассемблера // Микропроцессорные средства и системы. — 1985, — № 4, — С. 35.
6. Елламаев Р. А., Галузо Е. В., Голованов В. П. Огладочная система для однокристалльной микроЭВМ КМ1816ВЕ48 // Микропроцессорные средства и системы. — 1986, — № 3, — С. 32.

Статья поступила 22 сентября 1986 г.

УДК 681.327

А. С. Крянев, П. В. Белянкин

УСТРОЙСТВО СБОРА АНАЛОГОВОЙ ИНФОРМАЦИИ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Для анализа быстроменяющихся аналоговых сигналов необходимо накапливать следующие во времени отсчеты аналого-цифрового преобразователя (АЦП) в последовательных ячейках памяти. Известные способы ввода информации в микроЭВМ (программный и прямой доступ к памяти) не позволяют вести запись информации с высокой скоростью. Рационально процедуру обмена организовать путем накопления последовательности отсчетов от АЦП в быстродействующем буферном ОЗУ, минуя канал ЭВМ, и затем обращаться к этому ОЗУ как к массиву памяти в адресном пространстве микроЭВМ (рис. 1). Важной в этом случае оказывается доступность любых способов адресации, определенных для данного типа процессора при обработке информации. Обработка может выполняться без дополнительных перемещений данных, непосредственно в буферном ОЗУ.

Устройство, реализующее предложенный способ обмена, содержит встроенный АЦП и адресный счетчик, которые позволяют автономно накапливать последовательность отсчетов [1]. В адресном пространстве ЭВМ устройство представлено как массив памяти, положение которого определяется переключателями, установленными на плате, и имеет два последовательных адреса в банке внешних устройств: регистр состояния (РС) по адресу 170000₈ и регистр данных (РД) по адресу 170002₈.

Значения разрядов РС и РД приведены в таблице, форматы регистров — на рис. 2. Интерфейс устройства полностью согласован с каналом микроЭВМ «Электроника 60» [2]. Связь с каналом осуществляется через

Регистр	Разряд	Назначение	Вид доступа
РС	15	Устройство не готово	Считывание Запись-считывание Запись
	06	Разрешение прерывания	
	01	Пуск устройства	
РД	00—14	Начальный адрес массива данных	Запись

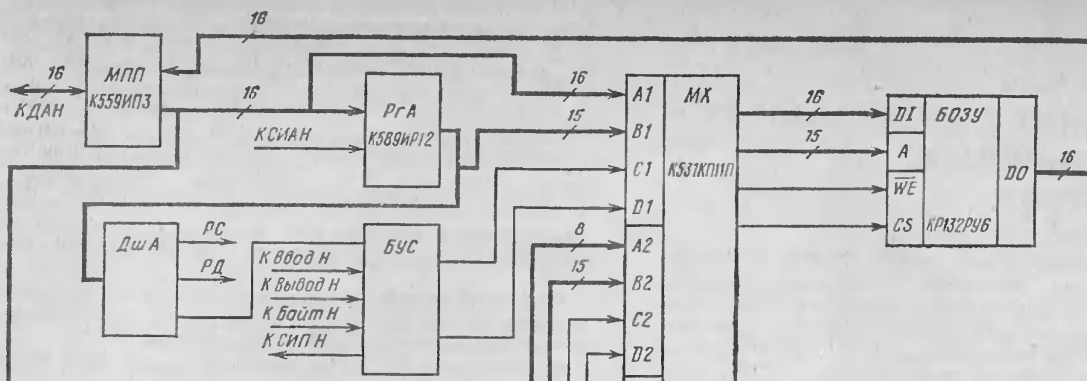


Рис. 1. Структурная схема устройства сбора аналоговой информации:

МПП — магистральный приемопередатчик; ДшА — дешифратор адреса; РГА — регистр адреса; БУС — блок управляющих сигналов; Тг — тактовый генератор; РС — регистр состояния; РД — регистр данных; БПр — блок прерывания; СЧА — счетчик адреса; БОЗУ — буферное ОЗУ

схем памяти КР132РУ6А, по оценке авторов, составляет 5 МГц. Реально блок эксплуатируется при частоте тактового генератора 2,5 МГц.

Ток потребления от источника питания напряжением 5 В не более 3 А, от источника —12 В — 0,4 А.

ЛИТЕРАТУРА

1. Горбачев С. Ф., Демин А. П. Оперативное запоминающее устройство с внешним скоростным каналом ввода информации в микроЭВМ «Электроника 60» // Микропроцессорные средства и системы.— 1986.— № 3.— С. 64—66.
2. Центральный процессор М2: Техническое описание и инструкция по эксплуатации.— М.: ЦНИИ «Электроника», 1979.
3. Микросхемы статического ОЗУ КР132РУ6 // Микропроцессорные средства и системы.— 1986.— № 3.— С. 92—94.

Адрес для получения консультаций: 188620, г. Пушкин Ленинградской обл., Ленинградский сельскохозяйственный институт, кафедра автоматизации сельскохозяйственного производства, тел. 470-05-65.

Сообщение поступило 27 октября 1986 г.

(Окончание. Начало см. на стр. 24)

Я бы мог раскрыть эти тезисы, но боюсь, что у кого-нибудь опять не хватит терпения, чтобы аккуратно разобратся в позиции автора (см. работы [2—5]).

А. А. Бабий

Мой адрес: 660001, Красноярск, ул. Менжинского, д. 12б, кв. 53.

ЛИТЕРАТУРА

1. Беляков Б. П., Зюляркин Д. Д., Подвальных А. М., Чернышев В. А. Производительные силы и производственные отношения в сфере производства программного продукта // Микропроцессорные средства и системы.— 1987.— № 2.
2. Бабий А. А. О некоторых характеристиках личной производительности труда программистов // УСИМ.— 1985.— № 6.
3. Бабий А. А., Быкова И. Н., Пискунова И. Ю. О СТП-оценке производительности труда программистов // Вычислительные методы и математическое моделирование. Тез. докл. Всесоюз. школы молодых ученых и специалистов.— Красноярск: 1986, с. 99—100.
4. Бабий А. А. Программирование как деятельность // Проблемы и опыт внедрения ЭВМ, АСУ и АСУ ТП. Тез. докл. кр. конф.— Красноярск: 1986, с. 45—51.
5. Бабий А. А. О термине «программное изделие» // Матер. 2-й Всесоюз. конф. по технологии программирования.— Киев, 1986.

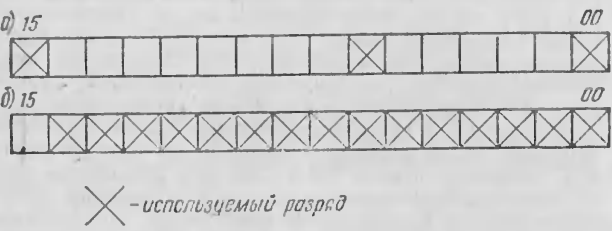


Рис. 2. Формат регистров состояния (а) и данных (б).

магистральные приемопередатчики. Буферное ОЗУ, имеющее организацию 16К×16 разрядов и выполненное на микросхемах статической памяти КР132РУ6 [3], позволяет обмениваться с микроЭВМ как словами, так и байтами и может работать в качестве стандартного ОЗУ.

Запуск процесса накопления производится записью единицы в младший разряд РС. Устройство применяется для работы с телекамерой «Волна 801», поэтому период отчетов задается встроенным тактовым генератором. В других случаях можно использовать внешнее тактирование. При заполнении последнего байта объема буферного ОЗУ процесс накопления отчетов автоматически завершается, т. е. сбрасывается старший бит РС. Если установлен разряд 6 РС, то возникает прерывание по вектору с адресом 270_в. Если число записываемых отчетов меньше максимально возможного объема буферного ОЗУ, равного 32К байт, то по адресу РД следует записать восьмеричное число, которое задает младший адрес массива накопления. При этом число отчетов будет определяться как разность между старшим адресом ОЗУ и записанным в РД числом.

Временные соотношения обмена с ЭВМ со стороны канала выдерживаются в соответствии с ГОСТ 26765.51-86. Со стороны аналогового входа максимальная частота квантования при использовании микро-

СОПРЯЖЕНИЕ ПЕРСОНАЛЬНЫХ ЭВМ С ПРИБОРНЫМ ИНТЕРФЕЙСОМ

Для эффективного использования ПЭВМ в физических экспериментах, связанных со сбором и обработкой измерительной информации, необходимо предусмотреть аппаратные и программные средства для поддержки приборного интерфейса [1, 2]. При наличии модулей «канала общего пользования» (КОП) полностью исключаются трудоемкие этапы ручного съема показаний с приборов и последующего ввода их с клавиатуры. Возможность быстрого создания экспериментальных установок гарантируется широкой номенклатурой серийно выпускаемых средств измерений с управлением от КОП (более 100 типов) и простотой сопряжения приборов с ЭВМ. Для эксперимента могут быть использованы почти все типы зарубежных приборов, так как стандарт на данный интерфейс является международным (МЭК 625.1, IEEE-488, HP-IB, IIS-2 и т. п.).

Во многих практических приложениях, когда не требуется высоких скоростей управления физическим процессом, целесообразно применение простейшего модуля КОП (так как между операциями обмена по КОП выполняются медленные операции — обработка данных), обеспечивающего только электрическое согласование двух магистралей: машинной и приборного интерфейса (рис. 1).

Программируемый интерфейс работает в режиме 0 — простой ввод-вывод. Назначение модуля КОП — чтение текущих состояний 16 линий и выдача уровней ТТЛ («Лог.0» или «Лог.1») с помощью портов В (управление шиной данных ЛД0...ЛД7) и С (шинами управления — линии КР, ДУ, ЗО, УП, ОИ и синхронизации — линии СД, ГП, ДП). Вывод информации в порт С осуществляется поразрядно с помощью управляющих байтов через порт Д. Для чтения состояний шин управления и синхронизации используется порт А, а шины данных — дополнительный порт Е, собранный на 2 микросхемах К589АП26. Требования по нагрузочной способности КОП ($U_{\text{вых}} < 0,5 \text{ В}$ при $I_{\text{вых}} = 48 \text{ мА}$) удовлетворяются за счет использования в выходном каскаде четырех микросхем К155ЛА13. Буферизация линий КОП осуществляется нагрузочными резисторами 3 и 6,2 кОм. Схема связи с внутримашинной магистралью зависит от типа ПЭВМ и должна обеспечивать выборку пяти адресов (порты А, В, С, Д, Е) при ис-

пользовании 8-разрядной ПЭВМ и 4 адресов (порты А и Е объединяются в один), если ПЭВМ 16-разрядная. Сигналы «запись» и «чтение» должны формироваться в соответствии с логикой работы внутримашинного интерфейса.

Если в ПЭВМ предусмотрена работа с периферийными устройствами через параллельный интерфейс или имеется цифровой ТТЛ ввод-вывод с числом входов и выходов не менее 16, а также осуществима независимая выдача уровней хотя бы по восьми линиям, то модуль КОП будет состоять только из выходного каскада.

Программное обеспечение для модуля КОП состоит из двух частей. К первой относится набор подпрограмм, выполняющих следующие функции:

выдачу уровней ТТЛ в любую из линий шин управления и синхронизации;

чтение состояний этих линий;

чтение и выдачу байта данных шине данных;

отработку функций синхронизации приема и передачи байта данных в соответствии с требованиями интерфейса КОП.

Программа драйвера на языке ассемблера для микропроцессора КР580ИК80 универсальна (текст написан с учетом отрицательной логики внутримашинного интерфейса, рис. 2). Настройка программы на конкретный тип ПЭВМ и модуля КОП занимает не более 10 мин. Драйвер обеспечивает скорость приема информации из КОП 3900 байт/с, скорость передачи 3800 байт/с (при частоте тактирования микропроцессора 2 МГц).

Вторая часть программного обеспечения содержит расширение языка высокого уровня, имеющегося в ПЭВМ, программами обработки группы операторов, ориентированных на КОП. В настоящее время отсутствуют нормативы и методические рекомендации по определению состава и назначения этих операторов. Существуют версии, включающие более 30 таких операторов (например, Бейсик ПМ в ИВК-М1), обеспечивающие любой режим работы КОП. Однако для реализации даже простых режимов пользователю необходимо в совершенстве владеть всеми требованиями информационной и программной совместимости интерфейса. Имеются расширения, состоящие из трех операторов (например, Бейсик GR в СДГ-01), они просты для пользователя-непрофессионала, но число режимов работы интерфейса в этом случае ограничено (отсутствуют важнейшие режимы: запуск прибора, опрос требующих обслуживания и др.). Поэтому разработка второй части программного обеспечения связана с определением состава и назначения операторов для поддержки модуля КОП. Необходимо, чтобы они были доступны для широкого круга пользователей и не урезали возможностей интерфейса.

Этим требованиям отвечает набор из 7 операторов: <TLK> [<#> <адрес> <,>] <список выражений, разделенных запятыми, или одно выражение> [<:>] <LSN> [<#> <адрес> <,>] <переменная любого типа> [<,> <выражение>] <SRQ> [<#> <адрес> <,>] <целая индексированная переменная> <LIN> [<список специальных записей типа 1, разделенных точкой с запятой, или одна специальная запись типа 1>] <AUT> <выражение> <FIG> <выражение> <PRI> [<список специальных записей типа 2, разделенных точкой с запятой, или одна специальная запись типа 2>]

Синтаксические диаграммы операндов: <адрес> и <выражение> — переменная или константа любого типа, допускаемая алгоритмическим языком ПЭВМ

<специальная запись типа 1> = <идентификатор> <0> или <идентификатор> <1> или <В> <,> <текстовая переменная или константа>

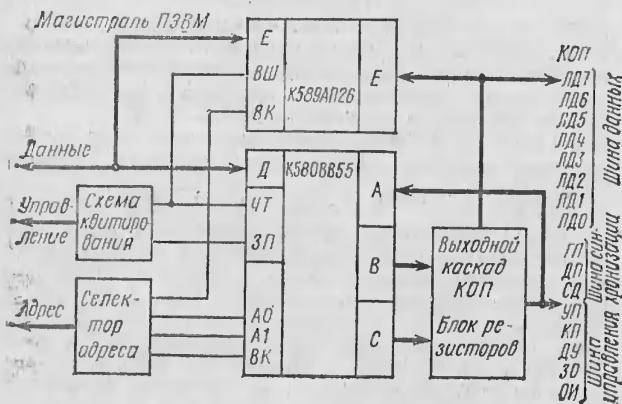


Рис. 1. Схема модуля КОП

```

; *****
; * УПРАВЛЯЮЩИЕ ПОДПРОГРАММЫ ДЛЯ МОДУЛЯ *
; * ИНТЕРФЕЙСА КОП ПЕРСОНАЛЬНОЙ ЭВМ *
; *****
; * ИСПОЛЬЗУЕМЫЕ РЕГИСТРЫ: * ОСНОВНЫЕ П/П: *
; * A,F,V,C,H,L * RB, RT, PRB, PRT *
; *****

```

ОПРЕДЕЛЕНИЯ ДЛЯ НАСТРОЙКИ ДРАЙВЕРА НА МОДУЛЬ КОП

```

RCP SET PORT1 ; ПРИ НАСТРОЙКЕ ПОЛЬЗОВАТЕЛЮ
RKP SET PORT2 ; НЕОБХОДИМО ЗАМЕНИТЬ ВЫРАЖЕНИЯ.
RDU SET PORT3
RZO SET PORT4 ; 1) PORT1...PORT9 НА
ROI SET PORT5 ; КОНКРЕТНЫЕ АДРЕСА ПОРТОВ
RSD SET PORT6 ; В/В, ПО КОТОРЫМ В ЕГО
RBP SET PORT7 ; МОДУЛЕ ОСУЩЕСТВЛЯЕТСЯ
RDP SET PORT8 ; ЧТЕНИЕ СОСТОЯНИЯ
RDATA SET PORT9 ; СООТВЕТСТВУЮЩИХ
SUP SET PORT10 ; ИНТЕРФЕЙСНЫХ ЛИНИЙ
SKP SET PORT11
SDU SET PORT12 ; 2) PORT10...PORT18
SZO SET PORT13 ; ВЫВОД УРОВНЕЙ "0" И "1"
SOI SET PORT14 ; В СООТВЕТСТВУЮЩИХ
SSD SET PORT15 ; ИНТЕРФЕЙСНЫЕ ЛИНИИ
SGP SET PORT16
SDP SET PORT17 ; 3) BAIT1...BAIT8 НА
SDATA SET PORT18 ; КОНТРОЛЬНЫЕ БАЙТЫ ДЛЯ
CUP SET BAIT1 ; КАЖДОЙ ЛИНИИ КОП, КРОМЕ ШД.
CKP SET BAIT2 ; КОНТРОЛЬНЫЙ БАЙТ ДОЛЖЕН
CDU SET BAIT3 ; ИМЕТЬ ЗНАЧЕНИЕ, ВОЗВРАЩАЕМОЕ
CZO SET BAIT4 ; ИЗ ПОРТА ВВОДА ПРИ АКТИВНОЙ
COI SET BAIT5 ; ЛИНИИ, ДЛЯ КОТОРОЙ ON
CSD SET BAIT6 ; ПРЕДНАЗНАЧЕН, НАПРИМЕР:
CGP SET BAIT7 ; "CUP SET 2H" (НИЗКОЕ
CDP SET BAIT8 ; СОСТОЯНИЕ ЛИНИИ "UP"
UPO SET BAIT9 ; ПЕРЕДАЕТСЯ ЕДИНИЦЕЙ ПО МА1
KPO SET BAIT10 ; В ЭВМ, А ВЫСОКОЕ - НУЛЕМ)
DUO SET BAIT11
ZOO SET BAIT12 ; 4) BAIT9...BAIT24 НА БАЙТЫ,
DIO SET BAIT13 ; ЗАПИСЬ КОТОРЫХ В
SDO SET BAIT14 ; СООТВЕТСТВУЮЩИЙ ПОРТ ВЫВОДА
GPO SET BAIT15 ; ОБЕСПЕЧИВАЕТ ПЕРЕВОД
DPO SET BAIT16 ; ВЫБРАННОЙ ЛИНИИ В НИЗКОЕ
UP1 SET BAIT17 ; (НАПРИМЕР "DUO") ИЛИ ВЫСОКОЕ
KP1 SET BAIT18 ; (НАПРИМЕР "DU1") СОСТОЯНИЕ
DU1 SET BAIT19
ZO1 SET BAIT20 ; 5) BAIT25 - ЗНАЧЕНИЕ ДЛЯ
OI1 SET BAIT21 ; УСТАНОВКИ ЛД0...ЛД7 В ВЫСОКОЕ.
SD1 SET BAIT22
GP1 SET BAIT23 ; 6) BAIT26 - ЛЮБОЙ КОД, НЕ
DP1 SET BAIT24 ; РАВНЫМ BAIT1...BAIT25
SUB SET BAIT25
SBO1 SET BAIT26

```

ОПРЕДЕЛЕНИЯ

```

RCFO MACRO LINE,CBAIT ; УСТАНОВКА БИТА НУЛЯ В
IN LINE ; (1), ЕСЛИ ЛИНИЯ "LINE"
ANI CBAIT ; В НИЗКОМ СОСТОЯНИИ
ENDM
RCF1 MACRO LINE,CBAIT ; УСТАНОВКА БИТА НУЛЯ В
IN LINE ; (1), ЕСЛИ ЛИНИЯ "LINE"
CMA ; В ВЫСОКОМ СОСТОЯНИИ
ANI CBAIT
ENDM
RBTD MACRO ; СОСТОЯНИЕ ШИНЫ ДАННЫХ
IN RDATA ; ЛД0...ЛД7 - В С-РЕГИСТР
CMA
MOV C,A
ENDM
STN MACRO LINE,CBAIT ; УСТАНОВКА ЛИНИИ "LINE"
MVI A,CBAIT ; В ВЫСОКОЕ ИЛИ НИЗКОЕ
OUT LINE ; СОСТОЯНИЕ
ENDM
SBTD MACRO ; УСТАНОВКА БАЙТА ДАННЫХ
MOV A,C ; НА ЛД0...ЛД7 ИЗ
CMA ; С-РЕГИСТРА
OUT SDATA
ENDM

```

```

RGD11 MACRO ; УСТАНОВКА БИТА НУЛЯ
LOCAL RNOT ; В (1), ЕСЛИ ЛИНИЯ
RCF1 RDP,CDF ; "GP" И "DP" В ВЫСОКОМ
JNZ RNOT
RCF1 RGP,CGP
ENDM
RNOT: NOP
ENDM

```

```

RGD10 MACRO ; УСТАНОВКА БИТА НУЛЯ
LOCAL RNOT ; В (1), ЕСЛИ ЛИНИЯ "GP"
RCF1 RGP,CGP ; В ВЫСОКОМ, А "DP" В
JNZ RNOT ; НИЗКОМ
RCFO RDP,CDF
ENDM
RNOT: NOP
ENDM

```

```

RGD01 MACRO ; УСТАНОВКА БИТА НУЛЯ
LOCAL RNOT ; В (1), ЕСЛИ ЛИНИЯ "GP"
RCFO RGP,CGP ; В НИЗКОМ, А "DP" В
JNZ RNOT ; ВЫСОКОМ
RCF1 RDP,CDF
ENDM
RNOT: NOP
ENDM

```

----- ВСПОМОГАТЕЛЬНЫЕ ПОДПРОГРАММЫ -----

```

TXTVK: ; П/П ВЫВОДА НА ДИСПЛЕЙ
DS 20 ; ТЕКСТА ИЗ ПАМЯТИ ЧЕРЕЗ
RET ; N-L ПАРУ
LDA TMA ; ИНИЦИАЛИЗАЦИЯ
STA ZL1 ; ПРОГРАММН. ЗАДЕРЖКИ,
XRA A ; ДЛИНА КОТОРОЙ ЗАДАЕТСЯ
STA ZL2 ; В ЯЧЕЙКЕ "TMA"
STA ZL3
RET

```

```

ZD: LDA ZL3 ; ОТРАБОТКА ПРОГРАММНОЙ
DCR A ; ЗАДЕРЖКИ, ПРИ ОБРАЩЕНИИ
STA ZL3 ; К П/П УСТАНОВЛИВАЕТСЯ:
RNZ ; - БИТ НУЛЯ (0), ЕСЛИ
LDA ZL2 ; КОЛИЧЕСТВО ОБРАЩЕНИИ К
DCR A ; НЕИ МЕНЬШЕ ЗАДАННОГО
STA ZL2 ; ЧИСЛА
RNZ ; - БИТ НУЛЯ (1), ЕСЛИ
LDA ZL1 ; КОЛИЧЕСТВО ОБРАЩЕНИИ К
DCR A ; НЕИ ДОСТИГЛО ЗАДАННОГО
STA ZL1 ; ЧИСЛА
RET

```

```

NPR: STN SGP,CPO ; ПОДГОТОВКА ЭВМ К ПРИЕМУ
STN SDP,DPO ; ИНФОРМАЦИИ ИЗ КОП. ЭТА
XRA A ; П/П ДОЛЖНА БЫТЬ ВЫЗВАНА
STA PRKP ; ДО ВЫЗОВА П/П "PRT", ПУ
MVI A,129 ; ПЕРЕД ПЕРЕВОДОМ ЛИНИИ
STA DLBUF ; "UP" В НИЗКОЕ
RET
KPR: STN SDP,DP1 ; СНЯТИЕ РЕЖИМА ПРИЕМА В
STN SGP,GP1 ; ЭВМ
RET

```

----- ТЕКСТЫ ДЛЯ ДИАГНОСТИКИ ПРИБОРОВ, РАБОЧИЕ ЯЧЕЙКИ -----

```

T01: DB 'ОТСУТСТ. ПОДТВЕРЖД. О ПРИЕМЕ БАЙТА',0
T02: DB 'ПРИБОР НЕ ПЕРЕШЕЛ В РЕЖИМ ПРИЕМА',0
T03: DB 'ОТСУТСТ. ПОДТВЕРЖД. О ГОТ. К ПРИЕМУ',0
T04: DB 'НЕТ СИГНАЛА О СОПРОВОЖДЕНИИ ДАННЫХ',0
T05: DB 'НЕ ЗАВЕРШЕН ЦИКЛ ПЕРЕДАЧИ',0
T06: DB 'КОНЕЦ ТЕКСТА НЕ ОБНАРУЖ. ЗА 128 Б.',0
TMA: DB 1 ; ДЛИНА ТАЙМ-АУТА (ПО УМОЛ. -10 С.)
ZL1: DB 0
ZL2: DB 0 ; ЯЧЕЙКИ ДЛЯ ОТРАБОТКИ ЗАДЕРЖКИ
ZL3: DB 0
POSL: DB 0AH ; КОД ОГРАНИЧИТЕЛЯ В ЦИКЛАХ ОБМЕНА
OGR: DB 0 ; ПРИЗНАК ОТМЕНЫ ОГРАНИЧИТЕЛЯ (1)
PRKP: DB 0 ; ПРИЗНАК ОЖИДАНИЯ "KP"=0 НА ПРИЕМЕ
DLBUF: DB 12H ; ТЕКУЩЕЕ КОЛ. ПРИНЯТЫХ Б-ТОВ В BUF
ADR: DB 0 ; ТЕКУЩИЙ АДРЕС ПРИБОРА
RCMD: DB 0 ; НЕТ ЗАПРЕТА НА УСТ-КУ "KP" В "0"
RPRI: DB 0 ; НЕТ ЗАПРЕТА НА ВЫПОЛНЕНИЕ П/П KPR
BUF: DS 256 ; БУФЕР ПРИЕМА-ПЕРЕДАЧИ ДЛЯ КОП

```

```

; ----- ОСНОВНЫЕ ПОДПРОГРАММЫ -----
; **** ПОДПРОГРАММА ПЕРЕДАЧИ ОДНОГО БАЙТА ****
;
PB:   RGD11      ; ПРОВЕРКА ГОТОВНОСТИ
      JZ PB6    ; ПРИБОРА К ОБМЕНУ
      SBTD
PB1:  CALL IZD  ; ПОДГОТОВКА К ТАЙМ-АУТУ
      RGD10    ; ОЖИДАНИЕ ГОТОВНОСТИ
      JZ PB2   ; ПРИБОРА К ПРИЕМУ БАЙТА
      CALL ZD
      JNZ PB1
      LXI H,T03 ; ТАЙМ-АУТ. СВОИ В НАЧАЛЕ
      JMP PB4   ; ЦИКЛА ПЕРЕДАЧИ
PB2:  STN SSD,SDO ; СОПРОВОЖДЕНИЕ ДАННЫХ
      CALL IZD  ; ПОДГОТОВКА К ТАЙМ-АУТУ
PB3:  RGD01      ; ОЖИДАНИЕ ПОДТВЕРЖДЕНИЯ Q
      JZ PB5   ; ПРИНЯТИИ БАЙТА
      CALL ZD
      JNZ PB3
      STN SSD,SD1 ; ТАЙМ-АУТ. СВОИ В
      LXI H,T01  ; КОНЦЕ ЦИКЛА ПЕРЕДАЧИ
PB4:  MVI C,SVB
      SBTD
PB44: CALL TXTVK ; РАСПЕЧАТКА ПРИЧИНЫ СВОЯ
      MVI A,SBOI
      RET      ; ВОЗВРАТ В СЛУЧАЕ СВОЯ
PB5:  STN SSD,SD1 ; СНЯТИЕ СОПРОВОЖД. ДАННЫХ
      MVI C,SVB
      SBTD
PB6:  RET      ; НОРМАЛЬНЫЙ ВОЗВРАТ
      LXI H,T02 ; СВОИ ПО ГОТОВН. К ОБМЕНУ
      JMP PB4
;
; **** ПОДПРОГРАММА ПЕРЕДАЧИ ТЕКСТА ИЗ БУФЕРА ****
; **** ПЕРЕД ОБРАЩЕНИЕМ ЗАГРУЗИТЬ В В-РЕГИСТР ****
; *****ДЛИНУ ПЕРЕДАВАЕМОГО ТЕКСТА *****
PT:   LXI H,BUF
PT0:  MOV C,H   ; ОЧЕРЕДНОЙ ПЕРЕДАВАЕМЫЙ
      INX H    ; БАЙТ - В С-РЕГИСТР
      MOV A,B
      CPI 1    ; ПОСЛЕДНИЙ ЛИ БАЙТ?
      JNZ PT1
      LDA OGR  ; ДА
      CPI 1    ; ПРОВЕРКА НАЛИЧИЯ ОТМЕНЫ
      JNZ PT1  ; ОГРАНИЧИТЕЛЯ И ЗАПРЕТА
      LDA RCMD ; НА УСТАНОВКУ "КР" В
      CPI 1    ; НИЗКОЕ
      JZ PT1
      STN SKP,KPO ; ЗАПРЕТА НЕТ
      CALL PB   ; ПЕРЕДАЧА ОЧЕРЕДН. БАЙТА
      CPI SBOI  ; ПРОВЕРКА НА СВОИ
      JZ PT3
      DCR B    ; ВСЕ ЛИ БАЙТЫ ПЕРЕДАНЫ?
      JNZ PT0
      LDA OGR  ; ДА
      CPI 1    ; ПРОВЕРКА НАЛИЧИЯ ОТМЕНЫ
      JZ PT2  ; ОГРАНИЧИТЕЛЯ И ЗАПРЕТА
      LDA RCMD ; НА УСТАНОВКУ "КР" В
      CPI 1    ; НИЗКОЕ
      JZ PT11
      STN SKP,KPO ; ЗАПРЕТА НЕТ
      LDA POSL  ; ПЕРЕДАЧА ОГРАНИЧИТЕЛЯ
      MOV C,A
      CALL PB
      CPI SBOI  ; ПРОВЕРКА НА СВОИ
      JZ PT3
      STN SKP,KP1 ; ПРОВЕРКА НА СВОИ
      RET
PT2:  STN SKP,KP1 ; НОРМАЛЬНЫЙ ВОЗВРАТ
      RET
PT3:  STN SKP,KP1 ; ВОЗВРАТ В СЛУЧАЕ СВОЯ
      MVI A,SBOI
      RET
;
; **** ПОДПРОГРАММА ПРИЕМА ОДНОГО БАЙТА ИЗ КОП ****
PRB:  STN SGP,GP1 ; ВЫДАЧА ГОТОВНОСТИ ЭВМ
      CALL IZD   ; ПОДГОТОВКА К ТАЙМ-АУТУ
PRB1: RCFO RSD,CSD ; ОЖИДАНИЕ СОПРОВОЖДЕНИЯ
      JZ PRB3
      CALL ZD
      JNZ PRB1
      STN SDP,DP1 ; ТАЙМ-АУТ.

```

```

LXI H,T04 ; СВОИ В НАЧАЛЕ ЦИКЛА
;
PRB3: JMP PB44 ; СОСТОЯНИЕ ПРИЕМ У ЭВМ
      STN SGP,GP0 ; АНАЛИЗ ЛИНИИ "КР"
      RCFO RKP,CKP
      JNZ PRB4
      MVI A,1 ; "КР" В НИЗКОМ. ПОСЛЕДНИЙ
      STA PRKP ; БАЙТ
PRB4: RDATA ; ПРИЕМ БАЙТА
      STN SDP,DP1 ; ОКОНЧАНИЕ ПРИЕМА БАЙТА
      CALL IZD ; ПОДГОТОВКА К ТАЙМ-АУТУ
PRB5: RCFO RSD,CSD ; ОЖИДАНИЕ ОКОНЧАНИЯ
      JZ PRB6 ; ЦИКЛА ПЕРЕДАЧИ
      CALL ZD
      JNZ PRB5
      STN SGP,GP1 ; ТАЙМ-АУТ.
      LXI H,T05 ; СВОИ В КОНЦЕ ЦИКЛА ПРИЕМА
      JMP PB44
PRB6: STN SDP,DP0 ; ЗАВЕРШЕН. ЦИКЛА ПРИЕМА
      RET ; НОРМАЛЬНЫЙ ВОЗВРАТ
;
; **** ПОДПРОГРАММА ПРИЕМА ТЕКСТА ИЗ КОП ****
; ** В В-РЕГИСТР МОЖЕТ БЫТЬ ЗАНЕСЕНО ОГРАНИЧЕНИЕ **
; **** НА ДЛИНУ ТЕКСТА. ЕСЛИ ТАКОГО ОГРАНИЧЕНИЯ ****
; ** НЕТ, ТО В В-РЕГИСТР НЕОБХОДИМО ОБНУЛИТЬ. ****
PRT:  LXI H,BUF
PRT0: CALL PRB ; ПРИЕМ БАЙТА С-РЕГИСТР
      CPI SBOI ; ПРОВЕРКА НА СВОИ
      JZ PRT3
      MOV M,C ; ОЧЕРЕДНОЙ БАЙТ - В БУФЕР
;
      INX H
      LDA PRKP ; АНАЛИЗ ЛИНИИ "КР"
      CPI 1
      JZ PRT1
      DCR B ; АНАЛИЗ КОЛ. ПРИНЯТЫХ БТ.
      JZ PRT1
      LDA DLBUF
      DCR A ; НЕ ПЕРЕПОЛНЕН ЛИ БУФЕР?
      STA DLBUF
      JZ PRT2
      LDA POSL ; НЕТ. АНАЛИЗ НА КОД
      CMP C ; ОГРАНИЧИТЕЛЯ
      JNZ PRT0
      LDA RPRI ; ПРОВЕРКА ЗАПРЕТА
      CPI 1 ; НА ИСПОЛЬЗОВАНИЕ "КР"
      RZ ; ВОЗВРАТ БЕЗ "ЭВМ В ИСК."
PRT1: CALL KPR
      RET ; НОРМАЛЬНЫЙ ВОЗВРАТ
PRT2: LXI H,T06 ; СВОИ ПО ПЕРЕП. БУФЕРА
      CALL TXTVK
PRT3: CALL KPR ; ЭВМ В ИСХОДНОЕ СОСТОЯНИЕ
      MVI A,SBOI
      RET ; ВОЗВРАТ В СЛУЧАЕ СВОЯ

```

Рис. 2. Программа драйвера на языке ассемблера

<идентификатор> = <U> или <K> или <D> или <S> или <P> или <C> или <Z> или <O>
 <специальная запись типа 2> = <идентификатор> <.>
 <целая переменная> или <.> <целая индексированная переменная>

Работа оператора TLK (передача прибору, имеющему адрес <адрес>, последовательности байт, содержащихся в <списке выражений...>:

<список выражений...> преобразуется в последовательность байт;

линии ДУ и УП переводятся в низкое состояние; с помощью подпрограммы PB (при использовании названной подпрограмм и рабочих ячеек подразумевается обращение к соответствующим определениям драйвера) в КОП передаются команды «не принимай» (код 3FH), «не передавай» (код 5FH) и «мой адрес приемника» (код в соответствии со значением <адрес> плюс 200);

линия УП переводится в высокое состояние; с помощью подпрограммы PT в КОП передается

имеющаяся последовательность байтов и, если среди операндов отсутствует (<);, ограничитель; одновременно с последним передаваемым байтом и только на время его передачи осуществляется перевод линии КП в низкое состояние (при отсутствии поля <адрес> обрабатывается только эта операция с условием запрета на передачу ограничителя из ячейки POSL и линия КП переводится в низкое состояние; обычное использование оператора TLK без <адреса> — выдача в КОП интерфейсных сообщений);

линия УП переводится в низкое состояние;

с помощью подпрограммы PB передается команда «не принимай»;

линии УП и ДУ переводятся в высокое состояние.

Работа оператора LSN (присл от прибора, имеющего адрес <адрес>, последовательности байтов (данных измерений):

линия УП переводится в низкое состояние;

с помощью подпрограммы PB в КОП передаются команды «не принимай», «не передавай» и «мой адрес источника» (код в соответствии со значением <адрес> плюс 40H);

выполняется подпрограмма NPR;

линия УП переводится в высокое состояние;

с помощью подпрограммы PRT выполняется прием байтов (поступающих от прибора) до передачи ограничителя либо достижения линией КП низкого состояния (одновременно с передачей байта), либо до принятия числа байтов указанных в <выражении> (при наличии поля <(<); <выражение>»);

линия УП переводится в низкое состояние;

с помощью подпрограммы PB в КОП передается команда «не передавай»;

линия УП переводится в высокое состояние;

принятая последовательность байтов возвращается значению <переменной...>, если в этом операнде указана текстовая переменная; иначе производится фильтрация принятых байтов с целью выделения цифрового значения, осуществляются преобразование цифрового значения во внутреннее представление и присвоение <переменной...>. Использование обычных переменных (не текстовых) значительно сокращает суммарное время работы программы.

После выполнения операции при отсутствии поля <адрес> вызов подпрограммы KPR не производится. Оператор LSN без <адреса> необходим для приема длинных сообщений по КОП.

Работа оператора SRQ (присл от прибора, имеющего адрес <адрес>, байта состояния):

линия УП переводится в низкое состояние;

с помощью подпрограммы PB в КОП передается команда «не принимай», «не передавай», «открытие последовательного опроса» (код 18H) и «мой адрес источника»;

выполняется подпрограмма NPR;

линия УП переводится в высокое состояние;

с помощью подпрограммы PRB осуществляется прием байта состояния;

линия УП переводится в низкое состояние;

выполняется подпрограмма KPR;

с помощью подпрограммы PB в КОП передаются команды «не передавай» и «запрос последовательного опроса» (код 19H);

линия УП переводится в высокое состояние;

значение младшего бита принятого байта возвращается <...переменной> с индексом 1, следующего бита — с индексом 2 и т. д. до 8; значение принятого байта в целом возвращается <...переменной> с индексом 9.

Оператор LIN управляет состояниями интерфейсных линий, <0> — перевод линии в низкое состояние, <1> — в высокое. Все линии КОП имеют условное обозначение, посылаемое в <идентификаторе> (<U> — УП, <K> — КП, <D> — ДУ, <S> — СД, <G> — ГП, <P> — ДП, <Z> —

ЗО, <O> — ОИ). Управление шиной данных осуществляется одновременно по всем восьми линиям с помощью идентификатора <В>, на шину данных выводится байт из <текстовой...>.

Оператор AUT устанавливает временную задержку на ожидание ответных действий прибора при обмене информацией по КОП. По умолчанию или при <выражении>=1 устанавливается задержка 10 с, при <выражении>=2 — 20 с и т. д. Если ответные действия прибора не проявляются за установленное время, выполнение программы прекращается, линии интерфейса приводятся в исходное состояние и распечатывается причина, вызвавшая сбой (см. текст драйвера). При выполнении оператора AUT в ячейку TMA посылается значение, равное <выражению>.

Оператор FIG определяет байт признака конца обмена информацией по КОП (ограничитель). Работа оператора FIG — загрузка ячейки POSL значением, равным <выражению>.

Оператор PRI читает состояния линий КОП. Если линия находится в низком состоянии, то <...переменной> возвращается значение 0, иначе — 1. Если указан идентификатор шины данных <В>, то в соответствующий массив, обозначенный <...переменной>, записываются девять значений (аналогично оператору SRQ).

Данный набор операторов является функционально полным для модуля КОП, ориентированного на ПЭВМ. В большинстве случаев достаточно использования наиболее простых операторов: TLK, LSN, SRQ, AUT, FIG, LIN U0, U1, O0, O1 и их спецификаций а также PRI Z, X.

Модуль КОП и программное обеспечение апробированы в течение двух лет на ЭВМ СО-04 (аналог СМ-1800). Программное обеспечение удобно пользователям и позволяет реализовывать любой режим работы в рамках функций СИ1, СП1, К1...К4, К5, К26 [1, 2]. Набор операторов рекомендуется в качестве основы для унификации программного обеспечения КОП.

Аппаратные средства для интерфейса КОП представляют собой простое устройство (4...10 микросхем), которое может быть изготовлено в лабораторных условиях.

Телефон для справок: 582-32-71, Москва

ЛИТЕРАТУРА

- ГОСТ 26.003-80 (СТ СЭВ 2740-80). Система интерфейса для измерительных устройств с байт последовательным, бит-параллельным обменом информацией. — М.: Стандарты, 1985.
- Гореликов Н. И., Домарацкий А. Н., Домарацкий С. Н. и др. Интерфейс для программируемых приборов в системах автоматизации эксперимента. — М.: Наука, 1981.

Статья поступила 10 сентября 1986 г.

ОТ РЕДАКЦИИ

Статья «Машинная графика» авторов Ю. М. Баяковского и В. А. Галактионова в № 4, 1987 г. (2-я и 3-я страницы обложки) представляет собой выдержку из опубликованного ими в 1986 г. одноименного популярного пособия для лекторов общества «Знание». Приведенные в нем снимки (цветные диапозитивы) фирм Digital Productions, Lucasfilm и Numerical Design, широко известны. В частности, некоторые из них были опубликованы в статье А. ван Дама («В мире науки», 1985, № 11 и сборник «Современный компьютер», — М.: Мир, 1986, перевод с англ. В. А. Галактионова).

А. В. Германенко, Г. М. Миньков

ПРИМЕНЕНИЕ МИКРОЭВМ «ИСКРА 1256» ДЛЯ ИССЛЕДОВАНИЯ БЫСТРО- ПРОТЕКАЮЩИХ ОДНОКРАТНЫХ ПРОЦЕССОВ

Для автоматизации экспериментов по исследованию однократных процессов кинетических явлений в полупроводниках в импульсных машинных полях, протекающих за время, меньшее времени ввода-вывода информации через встроеныные интерфейсные блоки микроЭВМ «Искра 1256», применяется устройство, предварительно запоминающее информацию и после окончания измерения переносящее ее в память ЭВМ с необходимой скоростью. Оно выполнено в виде отдельного блока и представляет быстродействующий аналого-цифровой преобразователь с запоминанием результата каждого измерения в цифровом виде. Информация переносится в аналоговой форме с использованием АЦП процессора (допустимая точность измерений — несколько процентов). При такой организации обмена — минимальное число соединительных проводов между устройством и ЭВМ — уменьшаются аппаратные затраты. Автономность блока позволяет в случае поломки ЭВМ не останавливать измерений, а, используя осциллограф и графопостроитель, производить обработку результатов вручную.

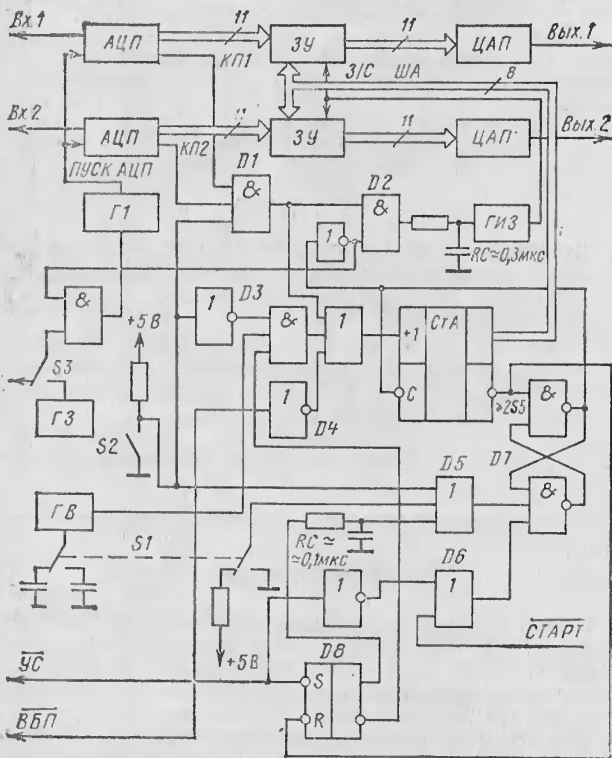


Рис. 1. Структурная схема устройства:

S1 — считывание на графопостроитель (←), осциллограф (→);
S2 — запись (←), считывание (→); S3 — внешний тактовый генератор (←), внутренний (→)

Технические характеристики

Диапазон входных напряжений, В	0...1
Количество двоичных разрядов	10 (плюс один знаковый)
Время преобразования, мкс	8
Минимальная длительность исследуемого сигнала (при количестве точек 256), мс	2,5

Устройство состоит из двух каналов регистрации информации (что расширяет его эксплуатационные возможности) и схемы управления (рис. 1). Каждый канал регистрации содержит АЦП типа Ф7077/1. Применение АЦП промышленного производства обеспечивает гарантированные метрологические характеристики при высоком быстродействии. Запоминающее устройство — 11 интегральных микросхем К527РУ2А, одноименные шины которых, за исключением информационных, соединены параллельно. Информационный вход каждой ИМС подключен к цифровому выходу АЦП, а информационный выход — к соответствующему разряду цифрового входа ЦАП. ЦАП выполнен на основе БИС К527ПА1 и операционного усилителя К140УД8.

Устройство управления состоит из адресного счетчика; генератора тактовых импульсов записи (ГЗ) с регулируемой частотой следования импульсов; генератора импульсов запуска АЦП (ГИЗ) длительностью 0,5 мкс; генератора импульсов воспроизведения (ГВ) с частотой следования 25 кГц для считывания информации на экран осциллографа и 5 Гц для вывода на графопостроитель в аварийном режиме; генератора импульсов записи в ЗУ ГИЗ длительностью 0,5 мкс и элементов D1...D8, обеспечивающих работоспособность схемы в режимах записи-воспроизведения и обмена информацией с микроЭВМ.

Режим записи. При поступлении сигнала СТАРТ триггер D7 принимает состояние, разрешающее работу адресному счетчику и прохождению импульсов с ГЗ либо внешнего задающего генератора (частота определяет скорость заполнения всех ячеек памяти) на генератор, формирующий сигнал ПУСК АЦП. Через 8 мкс (время преобразования АЦП) аналого-цифровой преобразователь формирует сигнал КП, по переднему фронту которого с задержкой 0,3 мкс и при наличии уровня разрешения на входе D2 генератором ГИЗ формируется импульс записи информации (считывания), выставленной к этому времени АЦП на входах ЗУ, в ячейку с текущим номером. По заднему фронту импульса КП происходит увеличение содержимого адресного счетчика на единицу, и процесс записи повторяется. При заполнении счетчика триггер D7 вырабатывает сигнал останова счета.

Для воспроизведения информации на вход счетчика подаются импульсы с ГВ. Работа триггера D7 блокируется уровнем «0» на одном из его входов, что обеспечивает периодическое (с частотой около 100 Гц) появление информации на экране осциллографа. Триггер D8 необходим для сброса счетчика в нуль и формирования уровня запрета прохождения импульсов счета с ГВ в процессе обмена информацией с ЭВМ. Сброс D8 происходит по сигналу УСТАНОВИТЬ СВЯЗЬ (УС). Содержимое счетчика в этом режиме увеличивается при поступлении сигнала ВВП. Сигналы УС и ВВП снима-

1 СЕЛЕКТ (1.0.10)	■ ЗАДАНИЕ 10 СИМВОЛЬНЫХ ПЕРЕМЕННЫХ ДЛИНОЙ 256 СИМВОЛОВ
2 П1	*
3 СЕЛЕКТ (12, 32, 0)	*
4 32 → И12	■ ПРОГРАММИРОВАНИЕ АЦП ИСКРА 015-11
5 11 → И14	■ НА ВВОД ПО 1-МУ КАНАЛУ
6 32 → И13	■ НА ПРЕДЕЛЕ 5В
7 F07	*
8 ВВОД И(1, 1, СA00, 2)	* ВВОД ПО 1-МУ КАНАЛУ 512 БАЙТ В СA00-СA03
9 32 → И13	* ПРОГРАММИРОВАНИЕ НА ВВОД ПО 2-МУ КАНАЛУ
10 F07	*
11 ВВОД И(1, 1, СA02, 2)	* ВВОД ПО 2-МУ КАНАЛУ 512 БАЙТ В СA02-СA03

Рис. 2. Программа ввода информации в память ЭВМ

ются с соответствующих выводов дешифратора команд АЦП «Искра 015-11». Программа ввода информации в память ЭВМ приведена на рис. 2. После выполнения подпрограммы П1 в символьных переменных СА00...СА03 (256 символов каждая) в формате Ф13 будет размещена информация, находящаяся в момент обмена в запоминающем устройстве.

Подобное решение задачи сбора аналоговой информации не является специфичным для микроЭВМ. При использовании в качестве высокоскоростного АЦП современные БИС АЦП параллельного действия можно существенно улучшить минимальную длительность исследуемого процесса.

Такая организация возможна также в системах, не имеющих высокоскоростных АЦП, например «Искра 226», «Электроника Д3-28», СМ 1800 и т. д.

Адрес для справок: 620083, Свердловск, Уральский госуниверситет, тел. 22-02-72

Статья поступила 9 апреля 1986 г.

УДК 681.322.012:681.325

Ю. П. Барметов, С. А. Боев, Ю. И. Евтеев

АНАЛИЗАТОР СИГНАЛОВ АКУСТИЧЕСКОЙ ЭМИССИИ НА ОСНОВЕ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ

Разработанная микропроцессорная система управления информационно-измерительных систем (ИИС) имеет многопроцессорную структуру (рис. 1), которая включает управляющую микроЭВМ «Электроника 60», три процессора на базе комплекта БИС К1804, блок умножения, регистры связи, буферную память. Обработываемые сигналы от пьезодатчиков через усилители, коммутатор аналоговых сигналов, аналого-цифровой преобразователь и входные регистры поступают в каналы процессоров. Промежуточные результаты обработки накапливаются в локальных ОЗУ процессоров, конечные — пересылаются в буферную память для дальнейшего использования в ЭВМ.

Последовательность операций, выполняемая процес-

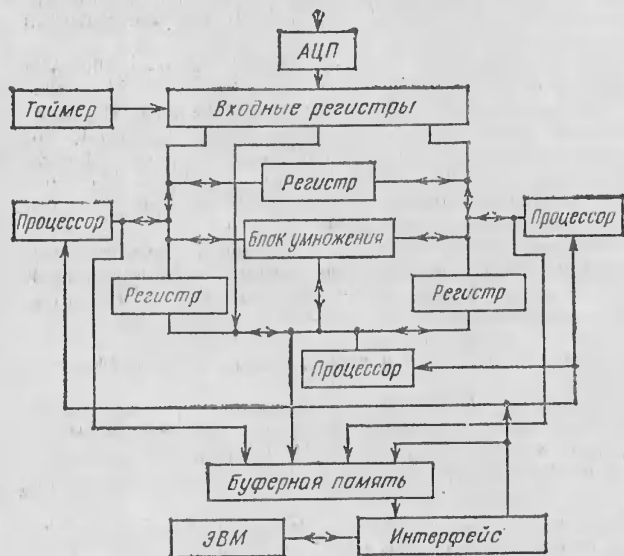


Рис. 1. Микропроцессорная система управления ИИС

сорами в ходе обработки сигналов, задается с помощью микроЭВМ — начальные адреса соответствующих программ загружаются в стековые области ОЗУ процессоров при подготовке системы к работе или после выполнения определенного ее этапа.

Основные операции, реализуемые процессорами: фильтрация, вычисление спектра, построение гистограмм, вычисление скользящего среднего и среднего квадратического отклонения. Различные процессоры могут иметь и разные наборы программ. Обмен данными между ними осуществляется через регистры связи К1804ИР3 в режиме прерывания программ.

Процессоры имеют одинаковую структуру на базе комплекта БИС К1804, ПЗУ К556РТ7, К541РУ2, ИС средней степени интеграции серий К555, К155, К531. Из комплекта К1804 используются микропроцессорные секции ВС1, схемы ускоренного переноса ВР1, схемы управления последовательностью микрокоманд ВУ4, регистры ИР1, ИР3.

Процессоры выполняют операции с фиксированной и плавающей запятой над числами нормальной и двойной длины, условные и безусловные переходы в программах, прерывания программ по запросам от ЭВМ и других процессоров, остановку программы при неготовности устройства источника или приемника операнда.

Отличие предложенной схемы от типовой — использование программных констант в поле команды. Для формирования команды загрузки регистра Q используется линейка схем И, ИЛИ. Ветвление программы, включающее проверку условия и изменение адреса команды, выполняется за один такт синхросигнала, увеличенный на 60 нс. Длительность такта сохраняется, если ветвления не происходит.

Процессор имеет управляемый генератор тактового сигнала (рис. 2) с программируемой длительностью такта 160, 200, 240 нс. Комбинированные команды перехода выполняются либо за один такт синхросигнала (если условие перехода не выполнено), либо за два;

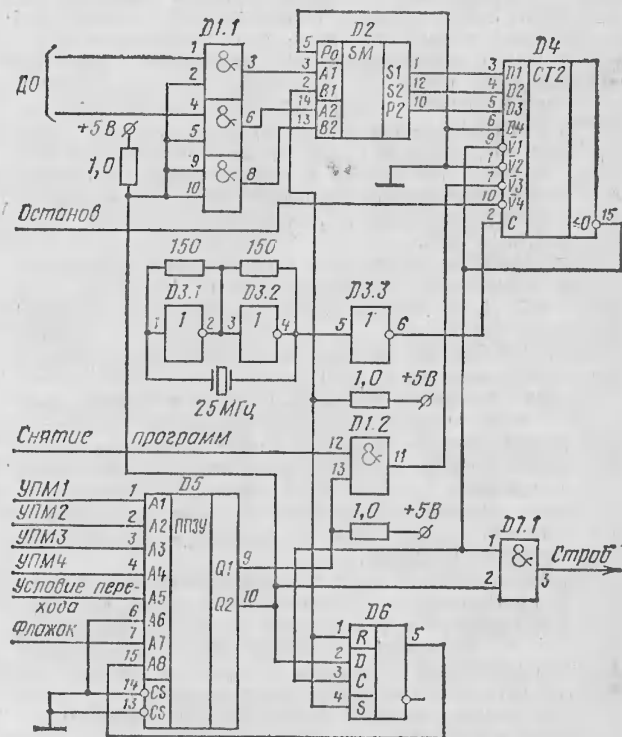


Рис. 2. Генератор тактового сигнала:

D1—К531ЛИ1; D2—К133ИМ2; D3—К531ЛН1; D4—К531ИЕ16; D5—К556РТ4; D6—К531ТМ2; D7—К531ЛИ1

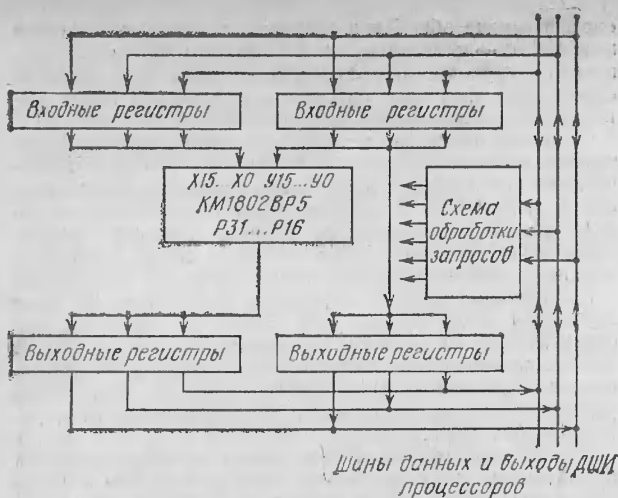


Рис. 3. Структурная схема блока умножения

длительность первого такта равна времени формирования условия перехода, второго — 80 нс.

Для исключения потери информации при обмене данными между блоком умножения, микроЭВМ и процессорами процессоры выполняют аппаратный анализ флажков состояния регистров-источников или регистров-приемников операндов. При неготовности регистра схема управления запрещает работу генератора тактового сигнала до изменения состояния флажка, т. е. останавливает выполнение программы в процессоре.

Операции умножения в системе реализуются на БИС KM1802BP5, длительность цикла которой для двух 16-разрядных слов не превышает 160 нс. При работе с 16-разрядным каналом процессора большую часть времени умножителя занимают операции обмена с регистрами, поэтому более рациональным является использование общего для всех трех процессоров блока умножения с промежуточными регистрами (рис. 3). Длительность операции умножения в разработанной системе по сравнению с выполнением аналогичной операции индивидуальным для каждого процессора умножителем возрастает в 1,1...1,7 раза (максимальное значение будет у процессора с самым низким приоритетом при одновременном обращении всех трех процессоров к блоку умножения) и составляет с учетом переключков 1,1...1,7 мкс.

Для отработки программного обеспечения применяются настраиваемые кросстрасляторы AMDASM или МИКРАС [1, 2]. Подготовлено программное обеспечение кросстраслятора для вычислительного комплекса 15ВУМС128-025 (на основе микроЭВМ «Электроника 60»), написанное на языке Паскаль.

Отладка программ осуществляется с помощью микроЭВМ и переходной платы-эмулятора или специального блока отладки, который также выполняет функции долгосрочного хранения неотлаженных программ, записи таблиц и готовых программ в ПЗУ. Микропроцессорная система анализатора сигналов акустической эмиссии используется при исследовании прочностных характеристик металлов.

Высокое быстродействие системы (до 15 млн. операций/с), программно-аппаратурная организация вычислительного процесса с реализацией операции умножения на БИС и применением таблиц тригонометрических функций позволяют проводить в реальном времени обработку сигналов для частот дискретизации до 1 МГц с привлечением операций цифровой фильтрации, спектрального и статистического анализа.

Адрес для справок: 394017, г. Воронеж, пр. Революции, д. 19, ВТИ, тел. 55-35-54

Статья поступила 30 октября 1986 г.

ЛИТЕРАТУРА

1. Проектирование цифровых систем на комплектах микропрограммируемых БИС/С. С. Булгаков, В. М. Мешеряков, В. В. Новоселов, Л. А. Шумилов.— М.: Радио и связь, 1984.
2. Микропроцессоры: системы программирования и отладки/В. А. Мясников, М. Б. Иснатъев, А. А. Кочкин, Ю. Е. Шейнин.— М.: Энергоатомиздат, 1985.

Статья поступила 30 октября 1986 г.

УДК 681.323:645.15

В. П. Тарасенко, Я. А. Рахлин, Я. И. Торшанко, Л. А. Савченко, И. П. Дробязко, Е. М. Швец

ИЗМЕРИТЕЛЬНО-ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС КОНТРОЛЯ ПАРАМЕТРОВ КАНАЛОВ ТОНАЛЬНОЙ ЧАСТОТЫ

Созданный на базе микропроцессора КР5801К80А измерительно-вычислительный комплекс АИСТ-ТЧ выполняет автоматические измерения и вычисления основных случайно изменяющихся параметров каналов тональной частоты: уровня приема, числа отказов, времени их действия, занижений и импульсных помех, скачков фазы и уровня, дрожания фазы и частоты, психометрического и невзвешенного шума. По результатам измерений определяются средние и экстремальные значения. Результаты измерений и статистической обработки отображаются на цифровом табло и документируются с помощью телеграфного аппарата.

Конструктивно комплекс выполнен в корпусе 480×280×580 мм; его основой является микроЭВМ, содержащая процессорный блок, ПЗУ объемом 16К байт КР556РТ5, ОЗУ объемом 32К байт К565РУ5, блоки обработки прерываний К589ИК14, сопряжения, панели управления.

МикроЭВМ взаимодействует с датчиками измерительных сигналов с помощью блока обработки прерываний, который по сигналам от датчиков формирует запросы, инициирующие процесс измерения и обработки информации в микроЭВМ. Блок позволяет обрабатывать 23 запроса (8 векторов).

Программное обеспечение АИСТ-ТЧ ориентировано на решение задач управления измерениями; обработки и представления информации; включает программы измерительных уровней, обслуживающие запросы прерывания, обработки, общего управления, сервисных функций, программы тестирования.

Комплекс может работать в автоматическом, ручном, дистанционном и проверочном режимах. Дистанционный режим предполагает выполнение контроля с помощью двух (ведущий-ведомый) или более комплексов, размещенных в различных точках контролируемого канала. Программа измерений и режим работы задаются оператором.

Применение АИСТ-ТЧ при проведении исследований каналов связи позволяет полностью автоматизировать процесс измерений, обработки и регистрации результатов и обеспечивает получение оперативной комплексной оценки состояния каналов.

Адрес для справок: 252110, г. Киев, ул. Соломенская, 3. КОНИИС. Телефон: 276-63-67

Сообщение поступило 10 февраля 1987 г.

Уважаемая редакция!

Расскажите, пожалуйста, о работе, назначении выводов и программировании контроллера клавиатуры и индикации КР580ВВ79.

Ю. Н. Сидоров, инженер (Пенза)

УДК 681.325.5

Ю. И. Торгов

ПРОГРАММИРУЕМЫЙ КОНТРОЛЛЕР КЛАВИАТУРЫ КР580ВВ79

Одна из часто встречающихся на практике задач — анализ состояния набора датчиков и отображение цифровой информации на индикаторных дисплеях. Примеры такого набора датчиков — это стандартная клавиатура ввода символической информации, игровые клавиатуры синтезаторов, датчики пожарной и охранной сигнализации, датчики предельных положений подвижных элементов конструкций, предельных режимов аппаратуры и т. п. Состояния датчиков анализируются микропроцессорными средствами за счет периодического ввода сигналов датчиков и программной обработки введенной информации. Обычно для этого требуется дополнительная аппаратура для ввода и предварительной обработки сигналов датчиков и существенные за-

траты процессорного времени для выполнения программ ввода и анализа информации. Если частота опроса датчиков большая, то эти затраты времени, как и обслуживание индикаторных дисплеев (требующих периодической регенерации выдаваемой на дисплей информации), существенно снижают быстродействие процессора при решении основных задач. Применяв контроллер КР580ВВ79, можно резко упростить аппаратную часть и разгрузить процессор для решения основных задач.

Программируемый контроллер клавиатуры КР580ВВ79 — многофункциональное многорежимное устройство. Хотя его основное назначение — анализ состояния матрицы двухпозиционных датчиков, сигнализация об изменении их состояния и выдача закодированных номеров сработавших датчиков, контроллер можно использовать в качестве строблируемого порта ввода, 8-уровневого регистра с организацией FIFO (первый записан, первый прочитан) и дополнительного набора регистров — ОЗУ с организацией 16 слов × 8 разрядов.

Содержимое регистрового ОЗУ может циклически выдаваться через порт вывода синхронно с выдачей управляющих сигналов для обслуживания индикаторных дисплеев.

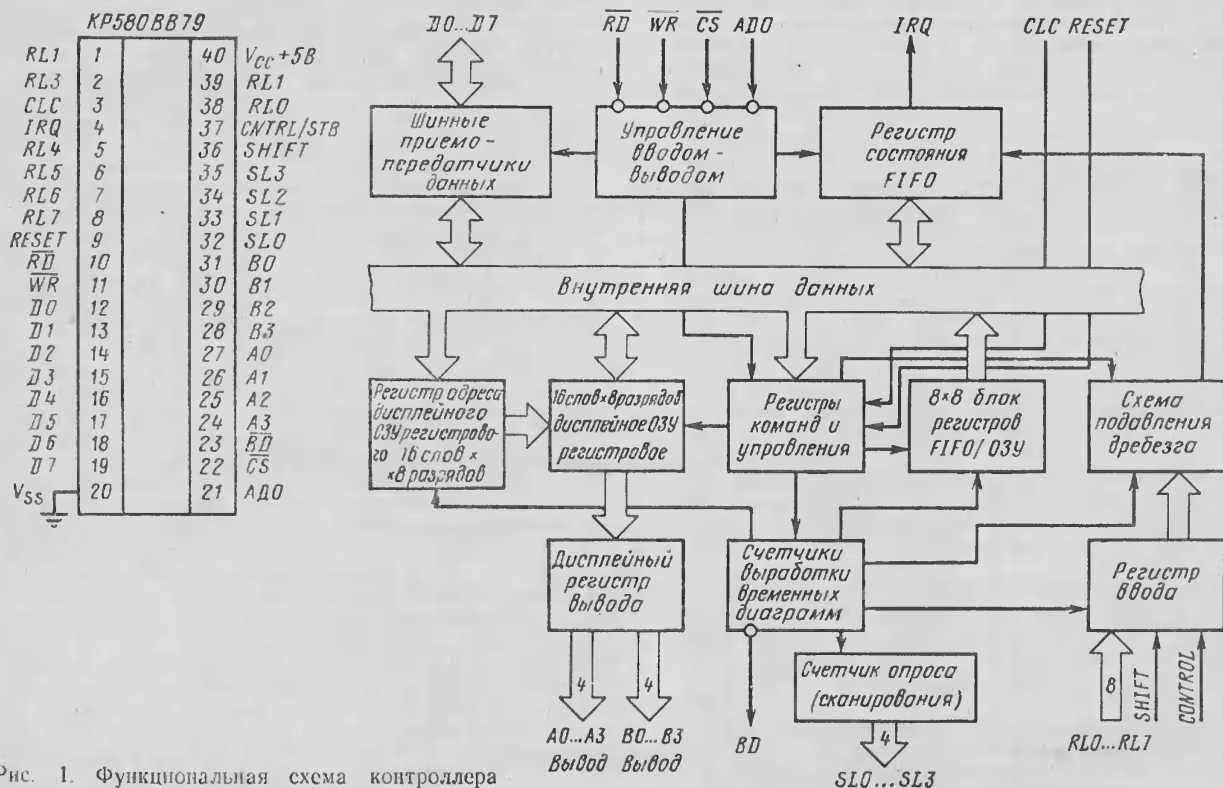


Рис. 1. Функциональная схема контроллера

Контроллер КР580ВВ79 рассчитан на работу под программным управлением микропроцессора, однако может работать и автономно, с аппаратной инициализацией после включения питания в нужный режим работы.

Принцип работы контроллера

Функциональная блок-схема контроллера и обозначение его выводов даны на рис. 1.

Блок связи с системной шиной использует стандартный для микропроцессора КР580ИК80 набор сигналов. Информация через двунаправленную 8-разрядную шину данных $D0...D7$ может быть записана или считана из внутренних регистров контроллера под управлением сигналов \overline{CS} , $A0$, \overline{RD} и \overline{WR} .

Тип информации определяется значением $A0$. Если сигнал на входе $A0$ — высокого уровня, то записываемая информация является кодом команды, а считываемая — кодом состояния (статус). Если сигнал на входе $A0$ — низкого уровня, то записываемая или считываемая информация является словом данных.

Обмен информацией возможен только при низком уровне сигнала \overline{CS} , в противном случае шинные формирователи контроллера находятся в состоянии высокого импеданса. Запись информации в контроллер происходит по условию $\overline{CS} \cdot \overline{WR}$, а считывание — по условию $\overline{CS} \cdot \overline{RD}$.

Блок регистров управления режимами работы контроллера и схема выработки временной диаграммы управляющих сигналов. Информация управления режимами обслуживания клавиатуры и дисплея заносится в регистры управления при записи в контроллер кодов команд ($A0=0$). Запоминание кода происходит по положительному фронту сигнала \overline{WR} , поэтому необходимо, чтобы сигнал \overline{WR} окончился ранее сигнала \overline{CS} при стабильном уровне сигналов $D0...D7$ на шине данных (см. рис. 2, б).

Временная диаграмма управляющих сигналов контроллера вырабатывается с помощью набора счетчиков. Первый счетчик — программируемый делитель частоты внешней синхросерии (CLC) с коэффициентом деления (2...31), устанавливаемым программно. Коэффициент деления, при котором базовая внутренняя частота

контроллера равна 100 кГц, дает время просмотра (сканирования) клавиатуры 5,15 мс и время ожидания успокоения дребезга контактов — 10,3 мс. Возможна работа и с пониженной базовой частотой с соответствующим изменением временных характеристик. С помощью остальных счетчиков из базовой внутренней частоты генерируются сигналы опроса (сканирования) клавиатуры и дисплейного ОЗУ. 4-разрядный счетчик сигналов опроса (сканирования) может работать в двух режимах — кодированном (обычного двоичного счета) и декодированном (режим сдвигового регистра с бегущим нулем).

В кодированном режиме для получения восьми сигналов опроса (сканирования) столбцов матрицы контактов размерностью 8×8 надо использовать внешний дешифратор «3 → 8», например КР155ИД7.

В декодированном режиме на выходе счетчика сканирования генерируются 4 сигнала опроса, так что необходимость в дополнительном дешифраторе отпадает, но размерность матрицы контактов не должна превышать $32 (8 \times 4)$.

Так как один и тот же счетчик вырабатывает сигналы опроса клавиатуры и включения дисплейных секций, то при работе счетчика в декодированном режиме будут работать только 4 разряда дисплея.

Входной 8-разрядный порт принимает и запоминает сигналы, приходящие со столбцов матрицы датчиков. Во всех режимах работы контроллера, кроме режима «ввод по стробу», информация вводится и запоминается в каждом такте работы счетчика опроса (сканирования). В режиме «ввод по стробу» код, поступивший на вход порта, запоминается в регистре FIFO по положительному фронту сигнала на входе «управление/строб» (CNTRL/STB).

Блок буферных регистров FIFO/ОЗУ контроллера работает в двух режимах. Если контроллер работает в режиме «опрос матрицы датчиков», блок регистров работает как нормальное ОЗУ формата 8 слов \times 8 разрядов. Каждое слово ОЗУ имеет свой адрес. Код со входа порта в буферное ОЗУ записывается циклически по адресу, равному номеру активного в этот момент сигнала опроса столбцов матрицы, вырабатываемому с помощью счетчика опроса (сканирования).

При обнаружении схемой анализа изменения состоя-

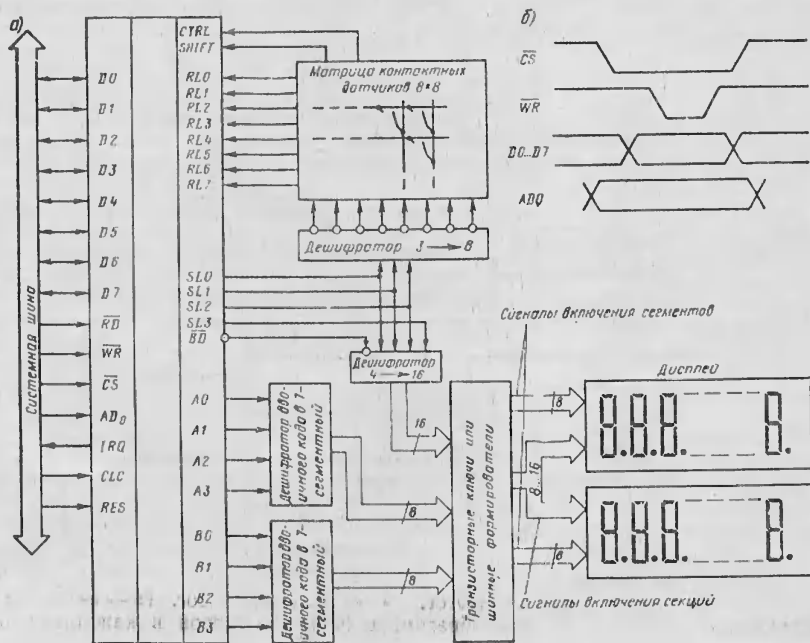


Рис. 2 Блок регистров управления режимами работы контроллера

ния какого-либо из датчиков по сравнению с предыдущим циклом запроса генерируется активный сигнал IRQ. Информация из ОЗУ считывается по адресу, устанавливаемому с помощью записи в контроллер специальной команды — «Чтение FIFO/ОЗУ датчиков».

В остальных режимах контроллера блок буферных регистров работает как 8-уровневый регистр FIFO (первый записан — первый считан). Коды автоматически записываются в последовательные позиции блока регистров, а безадресное считывание содержимого регистров происходит в порядке их записи. При считывании соответствующие регистры очищаются, позволяя записывать новую информацию в FIFO. В FIFO больше восьми слов информации не записывается. При попытке записи в него дополнительной информации устанавливается состояние «Лог. 1» разряда признака «переполнения» в слове состояния контроллера. Если число считываний из FIFO больше числа записей в него, то произойдет считывание недействительной информации из «пустого» FIFO. При этом будет установлен в «Лог. 1» разряд признака ошибки «Чтение пустого регистра». Если в FIFO имеется хотя бы один записанный и не считанный код (FIFO не пуст), то в слове состояния признак наличия информации в FIFO устанавливается в состояние «Лог. 1». При этом генерируется активный сигнал IRQ.

Схема подавления «дребезга контактов» (ПДК) работает в режимах «стандартной» и «игровой» клавиатуры. Она предотвращает распознавание кратковременных помех в качестве действительных срабатываний датчиков. Обнаруженное схемой анализа в цикле просмотра матрицы срабатывание датчика запускает схему ПДК. Эта схема запоминает номер сработавшего датчика, ждет в течение двух циклов просмотра и проверяет, остался ли датчик в сработавшем состоянии. Если да, то срабатывание — истинное, если нет — то оно ложное и игнорируется.

При внутренней частоте контроллера, равной 100 кГц, время выдержки схемы равно 10,3 мс. Таким образом, всякое срабатывание датчика на время, меньшее времени выдержки, будет игнорироваться. Программируя внутреннюю частоту контроллера, можно в широких пределах менять время просмотра матрицы датчиков и время выдержки ПДК применительно к рабочим характеристикам датчиков.

Схема анализа состояния матрицы датчиков в зависимости от режима работы контроллера анализирует условия срабатывания датчиков и управляет процессом записи информации в буферные регистры.

Дисплейное ОЗУ регенерации (16 слов × 8 разрядов) содержит информацию, выдаваемую через выходной порт на индикаторный дисплей. Эта информация записывается в ОЗУ регенерации программно через системную шину данных.

Регистр адреса ОЗУ регенерации содержит текущий адрес ячейки для обращения при записи или считывании слова данных. Код в регистр адреса заносится при записи в контроллер специальных команд: «Чтение ОЗУ регенерации» или «Запись в ОЗУ регенерации». Возможна установка режима с автоинкрементированием содержимого регистра адреса после каждого обращения к ОЗУ регенерации. Содержимое ОЗУ регенерации программно доступно процессору после установки соответствующего режима контроллера и адреса обращения.

Режимы анализа состояния датчика

Контроллер можно запрограммировать в одну из трех разновидностей режима анализа матрицы датчиков и вида выдаваемой информации о результатах анализа. Два режима («стандартной» и «игровой» клавиатуры) рассчитаны на то, что информация только о сработавших датчиках выдается в виде кода, индивидуального для каждого датчика в матрице. Коды выдаются в той последовательности, в которой схема анализа обнаруживает сработавшие датчики.

В режиме «матрица датчиков» информация о каждом из датчиков кодируется одним битом информации. В каждом цикле анализа матрицы информация о состоянии всей матрицы датчиков отображается в восьмиразрядных словах буферного ОЗУ в виде битовой карты.

В режиме стандартной клавиатуры схема анализа рассматривает в качестве действительного срабатывание только одного единственного датчика из всего набора. Если после обнаружения схемой анализа сработавшего датчика и запуска схемы ПДК на протяжении двух циклов просмотра матрицы (время работы схемы ПДК) других сработавших датчиков не обнаружено, то срабатывание датчика — единственное и код его позиции в матрице вместе с кодом состояния сигнала «регистр» (SHIFT) и управление (CTRL) заносится в блок регистров FIFO.

При этом вырабатывается сигнал наличия информации в блоке регистров — IRQ, если блок регистров до этого был пуст.

Если блок регистров был полностью загружен, то запись нового кода в регистр не производится, а будет установлен признак переполнения в слове состояния контроллера.

Если в течение времени работы схемы ПДК будут обнаружены другие сработавшие датчики, ввода информации в блок регистров не произойдет, пока все датчики, за исключением одного, не перейдут в несработавшее состояние. Последний оставшийся активным датчик будет рассматриваться как единственно сработавший, и информация о нем будет записана в блок регистров. Таким образом, запись информации о сработавшем датчике в блок регистров произойдет один раз за срабатывание, независимо от того, сколько датчиков сработало одновременно с ним и в каком порядке они прекращали срабатывание.

В режиме «игровой клавиатуры» каждое срабатывание датчика, обнаруженное при просмотре матрицы, рассматривается независимо от всех остальных. Если схема ПДК, работа которой описана выше, обнаруживает сработавший датчик на протяжении двух циклов опроса клавиатуры, то информация о нем заносится в блок регистров. Если одновременно сработало несколько датчиков, то информация о них заносится в блок регистров в том порядке, в котором схема анализа их обнаруживает.

Существует разновидность режима игровой клавиатуры, который устанавливается при записи в контроллер команды «снять прерывание/установить специальный режим анализа ошибки». При этом устанавливается флаг «Ошибка S/E» (SPECIAL ERROR) в слове состояния контроллера, если во время одного цикла работы схемы ПДК обнаруживаются два или более сработавших датчика. Установка этого флага прекращает запись в блок регистров новых кодов и устанавливает сигнал IRQ, если он не был установлен ранее. Ошибка может быть программно проанализирована после считывания слова состояния контроллера. Очистка флага производится командой «Очистка» с признаком $C_F=1$, после чего снова разрешается запись в блок регистров.

В режиме «Анализ матрицы датчиков» работа схемы подавления дребезга контактов запрещена. Информация о состоянии всех датчиков передается непосредственно в блок регистров, работающих как ОЗУ, в процессе просмотра матрицы.

Каждой строке матрицы датчиков прямо соответствует слово ОЗУ, в которое и записывается код, установленный при соответствующем сигнале просмотра на входе порта ввода. Состояния датчиков «управление» (CTRL) и «регистр» (SHIFT) — в этом режиме игнорируются.

Состоянию каждого датчика соответствует один разряд («Лог. 0» — выключен, «Лог. 1» — включен). Таким образом, в блоке регистров в каждом цикле оп-

доса хранится отображение всей матрицы датчиков в виде бит-карты.

Состояние датчиков можно анализировать программно, считывая содержимое ОЗУ. В частности, можно оценивать время нахождения датчиков во включенном состоянии. Сигнал прерывания (IRQ) устанавливается в «Лог. 1», если в конце цикла просмотра матрицы обнаружится, что хотя бы один датчик изменил состояние на обратное по сравнению с предыдущим циклом опроса. Дальнейшая запись в ОЗУ при этом запрещается.

Сигнал IRQ сбрасывается, и запись в ОЗУ разрешается при каждом считывании процессором данных из ОЗУ, если режим автономной адресации не установлен. В противном случае при записи в контроллер команды «Сбросить прерывание/режим специального анализа ошибки» сигнал прерывания сбрасывается.

Отметим, что срабатывание нескольких датчиков может вызвать многократные прерывания. В частности, это возможно при первом цикле анализа сразу после окончания сигнала «сброс» — (RESET).

В режиме «Ввод по стробу» информация вводится в буферный блок регистров FIFO со входов порта по положительному фронту сигнала на входе CTRL/STB. Данные могут поступать с датчиков или устройства, выдающего закодированную определенным образом информацию, в том числе с кодированной клавиатуры. Таким образом, в этом режиме контроллер может использоваться в качестве стробированного порта ввода с буфером FIFO на восемь слов.

Форматы информации в блоке регистров FIFO / буферном ОЗУ

В режимах «нормальной клавиатуры» и «игровой клавиатуры» информация о сработавшем датчике, записываемая в блок регистров, имеет следующий формат.

Шесть младших разрядов слова представляют собой код позиции датчика в матрице: разряды D0...D2 (R) — номер разряда входного порта, на котором принят сигнал датчика, а разряды D3...D6 (S) — номер сигнала опроса, при котором сигнал датчика был принят. Старшие два разряда отражают состояние двух дополнительных датчиков, подключенных непосредственно к входам порта «CONTROL» (D7) и «SHIFT» (D6). Сигналы CTRL (управление) и SHIFT (верхний регистр) на стандартной клавиатуре подаются с соответствующих клавиш.

Кодировка информации, считываемой из буферных регистров, не соответствует в большинстве случаев стандартам ГОСТ или ASCII. Если это необходимо, требуется программная или аппаратная перекодировка считанной информации.

В режиме «Анализ матрицы датчиков» сигналы датчиков с входов порта ввода записываются в соответствующие разряды слова ОЗУ, адрес которого соответствует номеру активного сигнала опроса. Формат кодового слова:

D7	D6	D5	D4	D3	D2	D1	D0
P17	P16	P15	P14	P13	P12	P11	P10

Кроме матрицы датчиков, к входу порта ввода можно подключать любые логические схемы, работающие под управлением сигналов опроса или сигналов счетчика опроса (например, до 8 мультиплексированных портов, состояние которых будет периодически анализироваться контроллером).

Использование сигнала IRQ

В режимах «нормальной» и «игровой» клавиатуры активный сигнал IRQ — признак того, что в буферных

регистрах есть информация, которую надо прочитать. Информация может быть считана в результате анализа сигнала IRQ при работе по флагу или по реакции на прерывание, если сигнал IRQ используется для работы с системой прерывания микропроцессора. При считывании слова данных из блока буферных регистров сигнал IRQ сбрасывается и снова становится активным, если в блоке регистров имеется еще непрочитанная информация. При работе контроллера в указанных режимах сигнал IRQ можно использовать в качестве сигнала «строб клавиатуры» или «готовность данных» (например, при автономной работе контроллера в составе клавиатуры, подключенной к микроЭВМ через порт, работающий в режиме стробируемого ввода).

В режимах «Анализ матрицы датчиков» и «Специальный режим анализа ошибки» появление сигнала IRQ указывает на изменение состояния датчиков или одновременное срабатывание двух и более датчиков соответственно.

В этих режимах целесообразно использовать сигнал IRQ в качестве сигнала прерывания для запуска программы анализа возникшей ситуации.

Слово состояния контроллера

Слово состояния контроллера содержит сведения о наличии записанной информации в FIFO, количестве записанных слов, возникновении ошибок переполнения или чтения пустого FIFO, одновременном срабатывании нескольких датчиков.

Формат слова состояния FIFO:

D7	D6	D5	D4	D3	D2	D1	D0
DN	S/E	0	F	F	N	N	N

где D2...D0 — количество записанных слов в FIFO; D3 — признак «FIFO» не пуст; D4 — признак «Чтение пустого «FIFO»»; D5 — признак «Запись в заполненный FIFO»; D6 (S/E) — в режиме «Анализ матрицы датчиков» обнаружено изменение состояния датчика, а в режиме «Специальный анализ ошибки» обнаружено одновременное срабатывание двух или более датчиков; D7 — дисплейное ОЗУ недоступно, так как выполняется команда «Очистить дисплейное ОЗУ» или «Очистить все»

Команды управления «клавиатурной» секцией контроллера

Команда «Установка режима работы клавиатурной и дисплейной секциями контроллера». Формат кода команды:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	D2	D1	K2	K1	C4

C4=0 — кодированный, и C4=1 — декодированный выходы счетчика сканирования,

K2 K1

0 0 — режим нормальной клавиатуры

0 1 — режим игровой клавиатуры

1 0 — режим анализа матрицы датчиков

1 1 — режим стробируемого ввода

D1=0 — вывод на 8 дисплейных секций

D1=1 — вывод на 16 дисплейных секций

D2=0 — ввод цифр слева

D2=1 — ввод цифр справа

Команда: «Установка коэффициента деления». Формат кода команды:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	P	P	P	P	P

Эта команда устанавливает коэффициент деления программируемого делителя частоты внешней синхросерии. При этом получается внутренняя синхросерия контроллера. При частоте, равной 100 кГц, период скарирования — 5,15 мс.

PPPPP может устанавливаться от 2 до 31.

После сигнала RESET-PPPPP=31.

Команда «Чтение FIFO/ОЗУ датчиков». Формат кода команды (X — значение разряда безразлично):

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	A1	X	A	A	A

Эта команда используется только при работе в режиме «Анализ матрицы датчиков». Она делает источ-

ником считываемых данных блок регистров FIFO, работающих в режиме ОЗУ.

Код AAA указывает номер регистра (адрес ячейки ОЗУ), из которого будет считана информация при очередном чтении данных.

Если A1=1 (автоинкремент), то после каждого чтения данных будет автоматически инкрементироваться код регистра адреса ОЗУ, и информация будет считываться из последовательных ячеек ОЗУ.

Если флаг A1=0, то чтение будет происходить из одной и той же ячейки ОЗУ до подачи новой команды с другим кодом адреса AAA.

Система обслуживания дисплея

Дисплейная часть контроллера рассчитана на обслуживание «мультиплексированных дисплейных секций без встроенной памяти». Пример — цифровой индикатор калькулятора. Каждая дисплейная секция имеет информационные входы и разрешающий вход. Информация на входах отображается в виде знака на индикаторе только при подаче разрешающего сигнала.

Необходимый для отображения знака код на информационных входах секции определяется типом дисплейной секции. В случае 7-сегментных индикаторов, используемых для отображения десятичных и шестнадцатеричных цифр, а также некоторых вспомогательных символов, на вход индикатора подается позиционный 8-разрядный код, где «Лог. 1» в каждом разряде кода соответствует включению соответствующего сегмента (7 сегментов+десятичная точка). В более сложных индикаторах, рассчитанных на отображение символьной

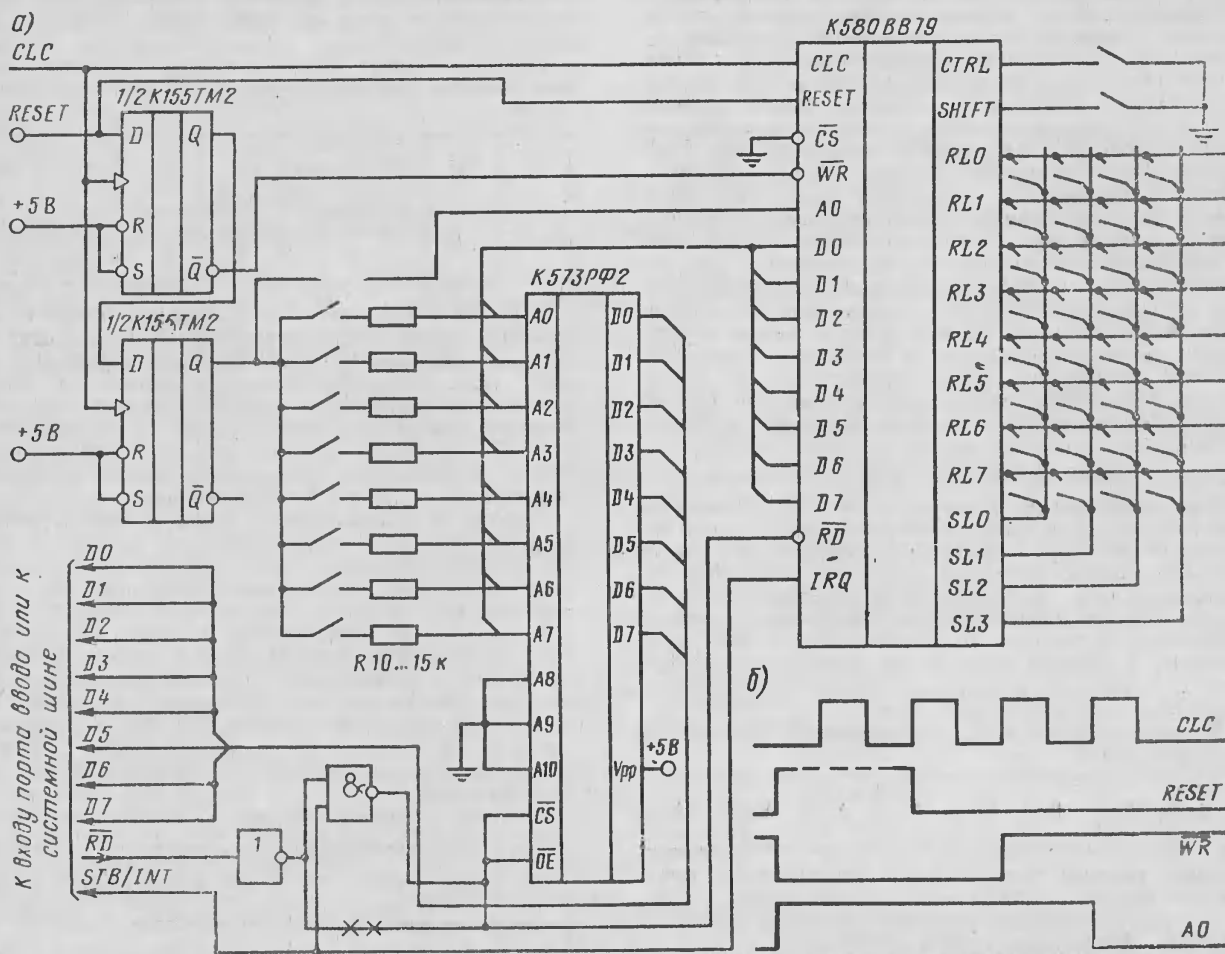


Рис. 3. Блок-схема дисплейной части контроллера

информации, обычно имеется внутренний шифратор, преобразующий входной стандартный код символа во внутренний код отображения.

Мультиплексирование дисплейных секций позволяет использовать один информационный канал для поочередного обслуживания всех дисплейных секций. Для этого код информации подается параллельно на входы всех дисплейных секций.

Циклический процесс поочередной выдачи кодов синхронизирован с последовательной выработкой разрешающих сигналов для каждой из секций. Хотя при этом в каждый момент времени работает только одна дисплейная секция, за счет инерционности зрения создается впечатление одновременной работы всех секций. Сами информационные коды хранятся в запоминающем устройстве небольшого объема (ОЗУ регенерации), циклически считываются оттуда и передаются на входы дисплейных секций через соответствующий порт вывода.

В блок-схему у дисплейной части контроллера (рис 3) входит ОЗУ регенерации объемом 16 слов по 8 бит (1), внутренние регистры адреса обращения к ОЗУ (2) и выходной разрядный порт данных (3). Сигналы общего с клавиатурой счетчика сканирования (4) управляют выборкой информации из ОЗУ и после каждого обращения инкрементируют код счетчика адреса. Это и обеспечивает циклическую выдачу содержимого ОЗУ регенерации через порт на входы дисплея.

В моменты изменения состояния счетчика сканирования вырабатывается сигнал гашения (БД) низкого уровня. Его можно использовать для гашения всего дисплея в моменты, когда информация на входах нестабильна, что вызывает появление заметных помех.

ОЗУ регенерации не связано аппаратно по данным с клавиатурной частью контроллера. Обмен данными с ОЗУ регенерации возможен только через системную шину данных под управлением процессора при выполнении команд обмена данными (A0=0) с контроллером. Информация в ячейки ОЗУ регенерации записывается процессором в такой кодовой форме, которая нужна на входе использованного дисплея. Информация в нужный вид перекодируется программно. Если на входе дисплея используется аппаратура перекодирования (например, КМ155ИД90 — кодировщик из шестнадцатеричного в семисегментный код) и можно использовать кодирование символа 4 разрядами вместо 8, то ОЗУ регенерации можно разделить на две параллельно работающие половинки с независимой записью информации в каждую из них. При этом количество дисплейных секций можно увеличить до 32.

Команды управления доступом к ОЗУ регенерации

Если процессор не обращается к ОЗУ регенерации контроллера, то источник адреса обращения — счетчик сканирования. При обращении со стороны процессора возможна только косвенная адресация ячеек ОЗУ регенерации, т. е. источником адреса становится внутренний регистр адреса. Поэтому необходимо сначала установить в регистре код адреса ячейки ОЗУ регенерации, в которую будет затем происходить обращение, т. е. записать в контроллер специальные команды установки кода адреса (при A0=1).

Команда «Запись в ОЗУ регенерации». Формат командного слова:

1	0	0	A1	A	A	A	A
---	---	---	----	---	---	---	---

Здесь старшие три разряда — код операции, младшие 4 разряда (AAAA) — код адреса ячейки ОЗУ, к которой произойдет следующее обращение процессора. A1 — флаг автономного режима.

Если A1=1, то после каждого обращения со стороны процессора в ячейке ОЗУ код регистра адреса

будет автоматически увеличиваться на 1. Это позволяет посылать команду только один раз, если необходимо записать информацию в последовательные ячейки ОЗУ регенерации.

Команда «Чтение из ОЗУ регенерации». Формат командного слова:

0	1	A	A1	A	A	A	A
---	---	---	----	---	---	---	---

Значения разрядов те же, что и в предыдущей команде. Эта команда устанавливает, помимо адреса ячейки, еще и внутренний переключатель, в результате чего источником считываемой информации становится ОЗУ регенерации, а не «буфер FIFO». Следует специально указать, что посылка команды «Запись в ОЗУ регенерации» не изменяет источника СЧИТЫВАЕМОЙ процессором информации. Таким образом, процессор будет читать информацию из источника определенного последней по времени командой («Чтение ОЗУ регенерации» или «Чтение буфера FIFO»). Однако вследствие того, что регистр адреса используется один и тот же при любом обращении процессора, как к ОЗУ регенерации, так и к «буферу FIFO», то код адреса в нем зависит от всей последовательности предыдущих действий. Это надо учитывать при составлении программ обмена информацией с контроллером. Во всяком случае, недопустимо перемежать различные виды обмена (запись и считывание, обращение к ОЗУ регенерации и буферу введенных символов) без предварительной посылки команды, устанавливающей код регистра адреса.

Команда «Запрет записи в ОЗУ регенерации, гашение дисплея (бланкирование)». Формат командного слова:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	X	Б. Зап. А	Б. Зап. Б	БЛН А	БЛН Б

«Лог. 1» в разрядах 2 и 3 командного слова запрещают запись кода соответственно в старший и младший полубайты ячейки ОЗУ. Это необходимо делать, если контроллер запрограммирован на работу с форматом дисплейного кода 2×4 разряда, обеспечивающим возможность включения до 2×16 дисплейных секций, и требуется изменить код только одного полубайта, не затрагивая содержимого второго полубайта, адресованного слова ОЗУ регенерации.

Разряды 0 и 1 команды — флаги гашения (бланкирования) дисплеев, подключенных к выходам А и В порта. Это позволяет гасить неиспользованную половину дисплейных секций, подавая на них код, при котором все сегменты индикаторов — темные. Вид кода зависит от типа дисплея и может быть 0000, 1111 или 00100000. Если контроллер запрограммирован на работу с 8-разрядным форматом дисплейного кода, то для гашения дисплея необходимо установить оба флага гашения. Сброс флагов гашения и включение дисплея выполняются при записи команды с нулевыми значениями соответствующих разрядов 0 и 1 командного слова.

Команда «Очистка». Формат командного слова:

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	ОД4	ОД3	ОД2	ОСБ	ОВ

Разряды ОД — управляют очисткой дисплейного ОЗУ регенерации.

Разряд ОД4, установленный в «Лог. 1», разрешает записать во все позиции ОЗУ регенерации один и тот же код, соответствующий погашенному дисплею (очистка дисплея). При этом комбинация разрядов ОД2 и ОД3 программирует, какой именно код будет записан:

ОД3	ОД2	
0	1	— все «Лог.0»
1	0	— 0010 0000 (20 ₁₆)
1	1	— все «Лог.1»

Операция очистки дисплейного ОЗУ занимает около 200 мкс. В это время старший разряд слова состояния (статус) установлен в «Лог. 1», и процессор должен анализировать состояние этого разряда после послышки команды «Очистка», прежде чем будет сделано новое обращение к контроллеру для записи новой информации. Разряд D1 (ОСБ) — управляет обнулением слова состояния (статуса) буфера введенных символов. После выполнения команды с установленным в «Лог. 1» разрядом ОСБ статус указывает, что буфер пуст, выходной сигнал прерывания сброшен, указатель адреса адресует нулевую ячейку буфера.

Разряд D0 — ОВ (очистить все) действует так же, как разряды D4 и D1 совместно. Кроме того, он перезапускает внутренний генератор тактовых сигналов, что может служить средством синхронизации.

Информация из ОЗУ регенерации на дисплей выдается синхронно с работой счетчика сканирования. Существует жесткое соответствие кода счетчика сканирования и адреса ячейки ОЗУ, из которой выдается информация. Так как код счетчика сканирования дешифруется для выработки сигналов включения дисплейных секций, то это обеспечивает отображение в каждой из секций содержимого одной и той же ячейки ОЗУ регенерации. При записи новой информации в ячейки ОЗУ регенерации, в зависимости от запрограммированного режима работы дисплейной части контроллера, адресация ячеек ОЗУ регенерации может смещаться по отношению к кодам счетчика сканирования. Визуально это воспринимается как смещение отображаемого на дисплее числа вправо или влево при появлении новых знаков.

В режиме заполнения дисплея со стороны старших разрядов (заполнение слева) каждому адресу ячейки в ОЗУ регенерации имеется прямое соответствие номера дисплейной секции.

Нулевой адрес ячейки ОЗУ соответствует старшему разряду дисплея, а адрес 15 (или 8 — при установке режима отображения 8 ячеек) — младшему разряду дисплея. Последовательная запись информации в ячейки ОЗУ регенерации с адреса 0 вызывает последовательное заполнение дисплейных секций со стороны старших разрядов. Когда после заполнения всех 16 (8) ячеек ОЗУ регенерации новая информация записывается опять в адрес ячейки 0, эта информация отображается в старшем разряде дисплея и т. д. (см. рис. 4).

Режим заполнения дисплея со стороны младших разрядов (заполнение справа). В большинстве калькуля-

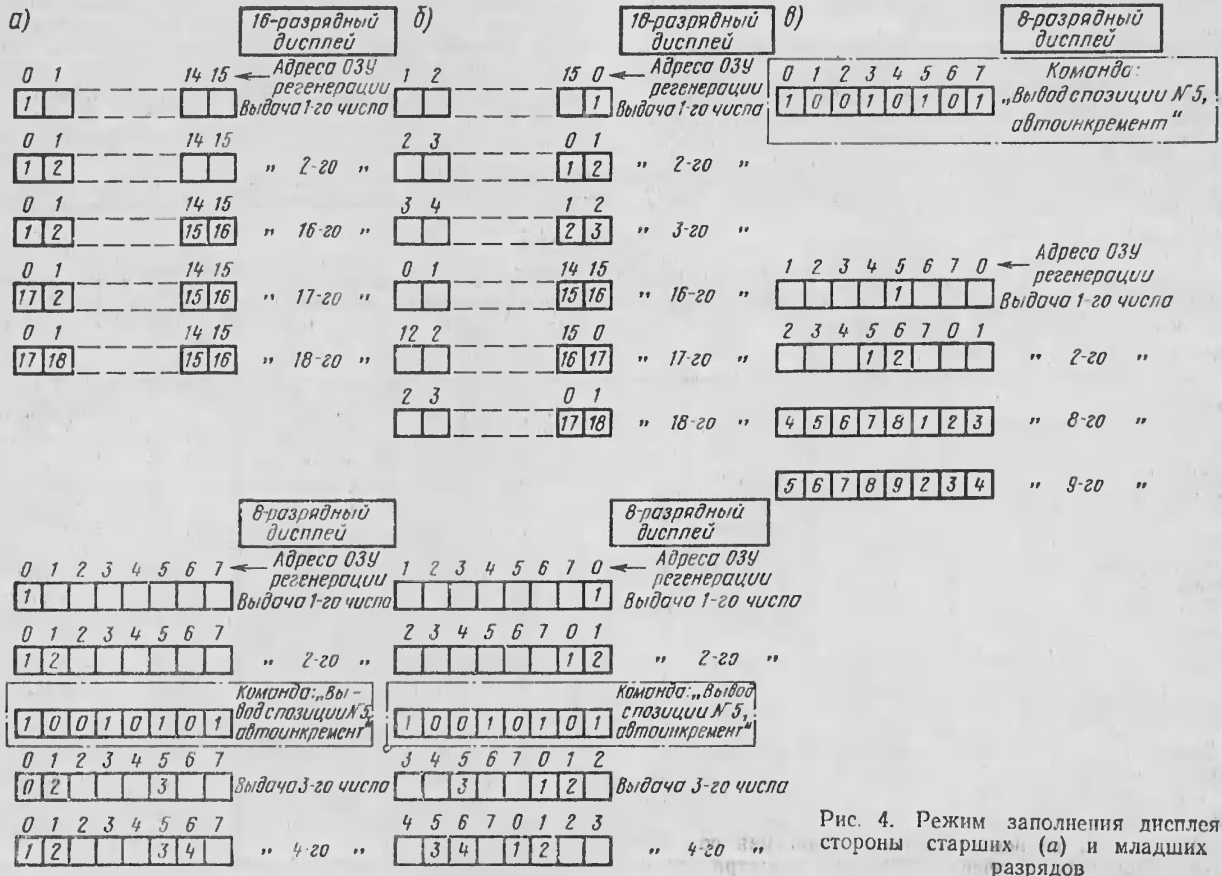


Рис. 4. Режим заполнения дисплея со стороны старших (а) и младших (б) разрядов

торов используется метод заполнения дисплея со стороны младших разрядов. Первая вводимая цифра помещается в младшем разряде дисплея, следующая — туда же, а ранее введенная цифра перемещается в сторону старших разрядов и т. д. (рис. 4, б).

Цифры, сдвигаемые из старшего разряда дисплея, теряются. При установке режима заполнения дисплея справа теряется прямое соответствие адресов ячеек ОЗУ регенерации и номеров дисплейных секций, в которых отображается информация.

При каждом вводе нового числа в ОЗУ в режиме автоинкрементации адреса код счетчика перемещается по отношению к адресам ОЗУ на единицу. Поэтому дисплей работает правильно после предварительной установки адреса на нулевую ячейку ОЗУ и при записи информации в автоинкрементном режиме.

Запись числа в произвольную ячейку ОЗУ при автоинкрементном режиме не рекомендуется, так как может вызвать непредсказуемый результат.

Режим автоинкрементирования адреса. В режиме

«заполнение справа» автоинкрементирование адреса ОЗУ при записи каждого нового числа последовательно заполняет ячейки ОЗУ и разряды дисплея. Без автоинкрементного режима запись ведется в одну и ту же ячейку ОЗУ и отображается в одном и том же разряде дисплея.

Это же справедливо и для режима «заполнения справа». Если запись в ОЗУ начать не с адреса 0, то заполнение дисплея начнется с разряда, соответствующего адресу ячейки ОЗУ, установленному командой «Запись в ОЗУ рег.». Новые разряды будут появляться именно в этой позиции дисплея, а ранее введенные будут сдвигаться по кольцу через старший разряд в нулевой разряд дисплея и пропадать при переходе к позиции, в которой появляются новые разряды (рис. 4, в).

Телефон для справок: 234-70-83. Адрес — 117333, г. Москва, ул. Вавилова, 40. ВЦ АН СССР. Статья поступила 7 августа 1986 г.

УДК 681.322.1+681.325.5

Д. И. Панфилов, С. Г. Шаронин, С. Е. Яковлев

СОПРЯЖЕНИЕ ЦАП И АЦП С МИКРОПРОЦЕССОРНЫМИ СИСТЕМАМИ

В зависимости от структуры, назначения, режимов работы и используемой элементной базы основные параметры современных микропроцессорных систем (МПС) изменяются в широком диапазоне. Эффективность использования МПС и возможностей ЦАП и АЦП зависит от способов их сопряжения.

Сопряжение ЦАП с микропроцессорными системами. При выборе способа сопряжения необходимо учитывать следующие характеристики ЦАП [1, 2]: разрядность,

логические уровни, требования к фронтам, метод преобразования информации из цифровой в аналоговую форму, скорость установления, точность и т. д. Основные характеристики наиболее распространенных ЦАП приведены в табл. 1.

Необходимым условием совместной работы ЦАП и МПС является их соответствие по видам цифрового кода и формата данных. В случае несовпадения кодов необходимо использовать внешние кодовые преобразователи. При сопряжении часто приходится решать задачу буферизации преобразуемого кода, так как не все ЦАП имеют внутренние регистры-фиксаторы для хранения полученных от МПС данных. На рис. 1 приведена схема, позволяющая благодаря наличию двойного ряда регистров-фиксаторов решать задачи, возникающие при сопряжении МПС с 8-разрядной магистралью данных и 12-разрядного ЦАП: буферизации данных, преобразования формата цифрового кода, исключения нежелательных переходных процессов при последователь-

Таблица 1

Характеристики интегральных ЦАП

Тип ЦАП	Число разрядов	Время установления выходного сигнала t_{ST} , мкс, не более	Дифференциальная нелинейность, %	Абсолютная погрешность преобразования в конечной точке шкалы, δ_{LD} , %	Диапазон опорного напряжения U_{REF} , В	Напряжение источников питания U_{CC} , В	Ток потребления I_{CC} , мА, не более	Технология
K127ПА1	15	30	$\pm 0,006$	$\pm 0,01$	± 10	$5 \pm 5\%$ $15 \pm 10\%$ $-15 \pm 10\%$	—	Гибридная
K572ПА1 (А...Г) КР572ПА1 (А...Г)	10	5	A $\pm 0,1$ B $\pm 0,2$ B $\pm 0,4$ Г $\pm 0,8$	$\pm 3,0$	± 17	5...17	2,0	ТТЛ, КМОП
K572ПА2 (А...В) КР572ПА (А...В)	12	15	A $\pm 0,025$ B $\pm 0,03$ B $\pm 0,1$	$\pm 0,5$	± 15	$5 \pm 5\%$ ($15 \pm 10\%$)	2,0	ТТЛ, КМОП
K594ПА1	12	3,5	$\pm 0,024$	—	$+10 \pm 3+$	5(5...15) $-15 \pm 10\%$	25,0 35,0	ТТЛ, КМОП
K1108ПА1 (А, Б)	12	A-0,4 B-0,7	$\pm 0,024$	$\pm 0,3$	2,2...10,5	$5 \pm 5\%$ $-15 \pm 10\%$	15,0 46,0	ТТЛ
K1118ПА1 K1118ПА2 (А...В)	8 A-10 B-9 B-8	0,02 0,03	+0,19 A $\pm 0,0978$ B $\pm 0,195$ B $\pm 0,391$	— $\pm 2,0$	+10 -1,0	-5,2+3% 5(1) $\pm 5\%$ -5(+5)%	130,0 15,0 120,0	ЭСЛ ТТЛ ЭСЛ

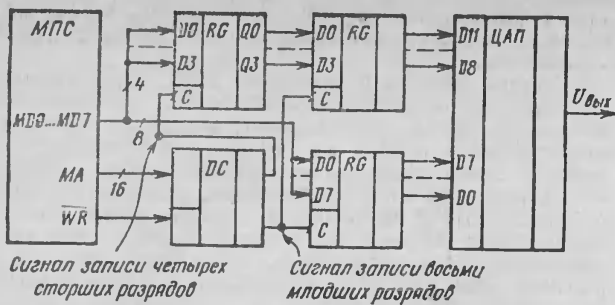


Рис. 1. Сопряжение 12-разрядного ЦАП и 8-разрядной МПС

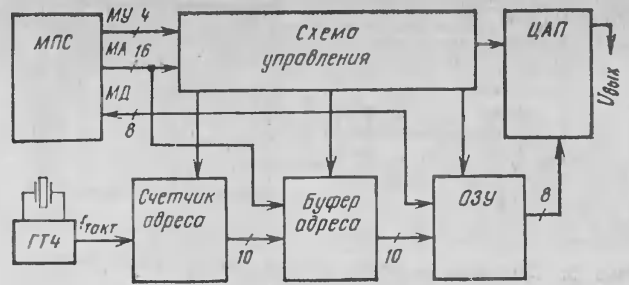


Рис. 2. Сопряжение ЦАП и МПС с использованием двухпортового ОЗУ

ной смене значений четырех старших и восьми младших разрядов. При этом минимальный шаг дискретизации по времени выходного аналогового сигнала ЦАП ограничен временем, за которое МПС выполняет два цикла записи данных в память или внешнее устройство.

Минимальное время установки быстродействующих ЦАП (20...40 нс) меньше, чем длительность цикла программного обмена данными между памятью и внешними устройствами для МПС широкого применения. Для согласования временных характеристик ЦАП и МПС, если требуется разрешающая способность по времени ЦАП большая, чем быстродействие МПС, используется обмен данными в режиме прямого доступа к памяти. При этом функции чтения данных из памяти и пересылки по адресу ЦАП выполняет не микропроцессорная БИС, а контроллер ПДП.

Быстродействие МПС зависит от времени выборки данных из памяти. Запоминающие устройства большой емкости и разрядности (ОЗУ серий К565, К581, К537; ПЗУ серий К573, К556, К558 и др.) имеют времена выборки, лежащие в пределах 200 нс...5 мкс [2]. Возможно сопряжение быстродействующих ЦАП с МПС невысокого быстродействия (например, с МПС, построенными на основе МКК серии КР580) (рис. 2). Для хранения данных, последовательно выводимых на ЦАП, отводится область памяти, схемотехнически реализуемая на быстродействующем двухпортовом ОЗУ (например, К132РУ4 с временем выборки 25 нс). Выводимые дан-

ные сначала записываются в ОЗУ, магистраль адреса (МА) МПС отключается с помощью буфера адреса и схемы управления, затем включается счетчик адреса и происходит последовательная выдача данных из ОЗУ на ЦАП с частотой, определяемой временем включения буфера адреса, временем выборки ОЗУ и временем установки ЦАП. Достоинства метода: практически полное использование быстродействия ЦАП; возможность периодической и однократной генерации сигнала; отсутствие потерь в быстродействии при подключении ЦАП с разрядностью, большей, чем длина слова МПС (это влияет на время записи данных в двухпортовое ОЗУ); малые потери в производительности МПС, так как во время генерации сигнала МПС может выполнять другую программу.

Сопряжение АЦП с микропроцессорными системами. Основные характеристики, определяющие метод подключения АЦП к МПС (табл. 2):

быстродействие АЦП (зависит, в основном, от способа преобразования аналогового сигнала в цифровую форму);

длина слова АЦП по сравнению с разрядностью магистрали данных МПС;

сигналы управления и состояния АЦП.

Основные сигналы состояния и управления АЦП: ЗАПУСК инициирует начало преобразования входного аналогового сигнала в цифровую форму; КОНЕЦ ПРЕОБРАЗОВАНИЯ (КП) указывает на завершение про-

Таблица 2

Характеристики интегральных АЦП широкого применения

Тип АЦП	Число разрядов	Нелинейность δL , % (или МЭР)	Дифференциальная нелинейность δLD , % (или МЭР)	Абсолютная погрешность преобразования в конечной точке шкалы δFS , % (или МЭР)	Время преобразования	Напряжение источника питания U_{CC} , В	Опорное напряжение U_{REF} , В	Диапазон входного напряжения U_{IN} , В	Токи потребления I_{CC} , А	Технология
К572ПВ1 (А...В)	12	0,00976 (А) 0,0976 (Б) 0,3506 (В)	0,0488 (А) 0,0976 (Б) 0,1953 (В)	(±3)	110...170 мкс	5±5% 15±10%	±15	0...- U_{an}	3 (U_{CC1}) 5 (U_{CC2})	ГТЛ КМОП
К572ПВ1 (А...В)	3,5 декады			(±1) (±3) (±5)	80...400 мс	5±5% -5±5%	±5	±5	1,8 1,8	КМОП
К1113ПВ1 (А...В)	10	±0,1 (А) ±0,2 (Б) ±0,4 (В) ±0,781	±0,1 (А) ±0,2 (Б) ±0,4 (В) ±0,781	(±4)	30 мкс	5±5% -15±5%	+10	-5...+5 0...10	10 20	ГТЛ
К1107ПВ1	6			±0,1	0,1 мкс	5±1% -1±1%	-0,075...0	-2,0...0	30 150	ГТЛ
К1107ПВ2	8	(±1/2)	(±1/2)		0,1 мкс	5±1% -1±1%	-0,075...0	-2,0...0	35 450	ГТЛ
К1107ПВ3	6	(±1/4)	(±1/4)		20 мс	5±1% -5,2±1%	-1,9...-2,1 +0,1		60 (U_{CC1}) 80 (U_{CC2}) 60 (U_{REF1}) 60 (U_{REF2})	ЭСЛ
К1108ПВ1	10 (8)	(±0,75)	(±1)	(±8)	0,9 (0,5) мкс	5±5% -5,2±3%	4,096	-0,1...4,2	50 90	ГТЛ

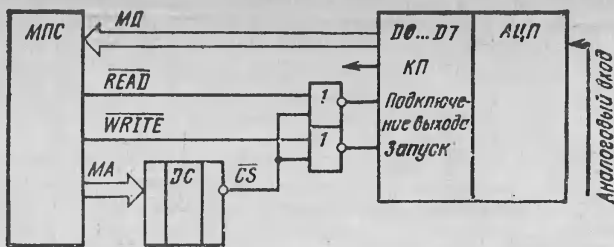


Рис. 3. Сопряжение 8-разрядного быстродействующего АЦП и МПС

цесса преобразования; ПОДКЛЮЧЕНИЕ ВЫХОДА (ПВ) позволяет выдачу информации (необходим для АЦП, имеющих цифровые выходы с тремя состояниями).

В дальнейшем будем полагать, что все АЦП имеют либо внутренние регистры, позволяющие запомнить код преобразованного аналогового сигнала и схемы с тремя состояниями, либо внешнее схемотехническое обеспечение, реализующее эти функции, а термином «быстродействующий АЦП» или «медленный АЦП» определять относительную скорость преобразования информации в АЦП по сравнению с временем выполнения команд в МПС.

Для быстрых АЦП наиболее простое подключение АЦП к МПС возможно, если время преобразования меньше времени выполнения команды (рис. 3). МПС подает сигнал на вход ЗАПУСК АЦП, используя операцию записи, затем считывает преобразованные данные с АЦП. Если время преобразования АЦП несколько больше, чем время выполнения команды, в программе, между командами запуска АЦП и считывания с него данных, нужно организовать выполнение цикла задержки на время, необходимое для выполнения процесса преобразования.

Для остановки работы МПС на время преобразования можно использовать сигнал КП (рис. 4), для получения результата используется команда чтения, во время которой формируются сигналы ЗАПУСК и ПВ. Использование одновибратора в схеме необходимо при подаче на выход ЗАПУСК импульса, а не логического уровня сигнала.

С момента начала процесса преобразования сигнала на выходе КП АЦП появляется сигнал, указывающий на наличие этого процесса. Выход КП непосредственно соединен со входом WAIT МП БИС, что обеспечивает режим остановки работы МПС на время преобразования. Время установки сигнала КП должно быть достаточным для того, чтобы во время выполнения МПС команды чтения успеть перевести его в режим ожидания. Преимущество такого подключения АЦП к МПС — выполнение одной команды чтения для запуска АЦП и ввода данных в МПС. Данные поступают в МПС за наиболее короткий промежуток времени, определяемый временем преобразования АЦП. При выборе АЦП для

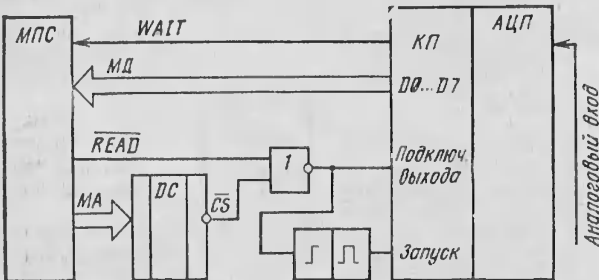


Рис. 4. Сопряжение быстродействующего АЦП и МПС с использованием сигнала КП

такого подключения необходимо учитывать, что время нахождения в состоянии ожидания некоторых МП БИС ограничено.

Применяя способы подключения АЦП, аналогичные приведенному на рис. 3, можно достичь максимально возможного числа преобразований в единицу времени, сократив число команд до трех, составляющих цикл выборки одного значения: считывание данных с АЦП, запись данных в стек, зацикливание. Дешифратор при появлении на MA МПС адреса команды записи в стек осуществляет запуск АЦП, а при заполнении стека необходимым количеством выборок формирует сигнал прерывания. Для МП БИС КР580ИК80 описанный цикл составляют команды MOVA, M; PUSH PSW и JMP, а суммарная длительность цикла (при тактовой частоте 2 МГц) — 14 мкс.

Для обеспечения частоты выборок, большей, чем позволяют описанные методы сопряжения, как и при подключении быстродействующих ЦАП, используется двухпортовое быстродействующее ОЗУ. Подключение выполняется аналогично рис. 2 с заменой блока ЦАП на блок АЦП и изменением направления соответствующей функциональной связи.

Для медленных АЦП использование описанных методов нерационально и приводит к потере эффективности использования рабочего времени МПС.

Передача данных по сигналам прерывания осуществляется специальной программой (рис. 5) с учетом временных диаграмм сигналов управления АЦП (рис. 6). При готовности данных АЦП к вводу, для формирования запроса прерывания используется сигнал КП с выхода АЦП. При выполнении подпрограммы обслуживания прерывания данные с выхода АЦП вводятся в МПС и инициируется следующий цикл преобразования.

Преимуществом такой организации сопряжения АЦП и МПС является простота: ввод данных осуществляется сразу же после преобразования в АЦП. Недостатком — сложность программного обеспечения, позволяющего в правильной последовательности организовать запуск АЦП, получение и отработку сигналов запроса прерывания и цифровых данных.

Обмен данными с помощью программного управления применяется, если немедленное введение данных с АЦП в МПС не является необходимым и используется программное управление обменом (рис. 7). После запуска АЦП МПС осуществляет проверку состояния АЦП, считывая непрерывно или через определенные промежутки времени состояние сигнала

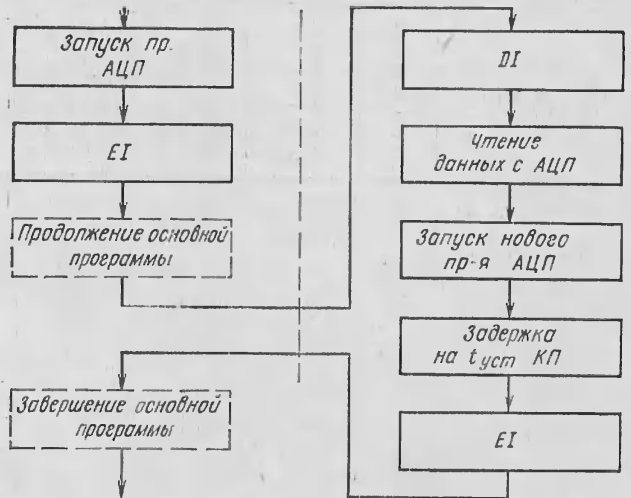


Рис. 5. Алгоритм программы обмена данными МПС с АЦП по сигналу прерывания

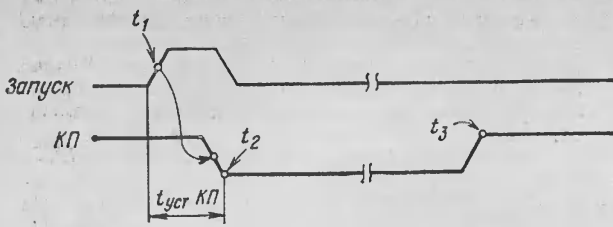


Рис. 6. Временные диаграммы сигналов управления АЦП при обмене по сигналу прерывания;

t_1 — начало преобразования; $t_{уст.КП}$ — время, в течение которого сигнал прерывания от предыдущего цикла может воздействовать на МПС; t_2 — только с этого момента должна быть осуществлена возможность восприятия сигнала прерывания; t_3 — запрос прерывания

КП (он подключен к разряду D7 магистрали данных МПС). По результатам проверки МПС либо считывает данные АЦП, либо продолжает выполнение программы ожидания.

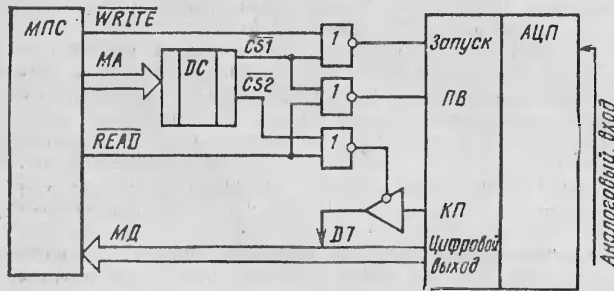


Рис. 7. Сопряжение АЦП с МПС при обмене данными с помощью программного управления

Подключение АЦП с использованием двойной записи данных (рис. 8) позволяет исключить в программе цикл ожидания готовности данных к вводу в МПС. Достигается это благодаря использованию дополнительного стробируемого регистра, из которого во время текущего цикла чтения считываются данные, полученные в предыдущем цикле преобразования. Для того чтобы данные в регистре во время выполнения команды чтения были неизменны, их запись в регистр (по переднему фронту импульса КП) и одновременно запуск следующего цикла преобразования АЦП осуществляются в те моменты времени, когда МПС не считывает содержимое регистра.

Программная реализация АЦП применяется при преобразовании медленно изменяющихся аналоговых сигналов с частотой не более тысячи раз в секунду и точностью 0,2—1,0%. Использование интегральных

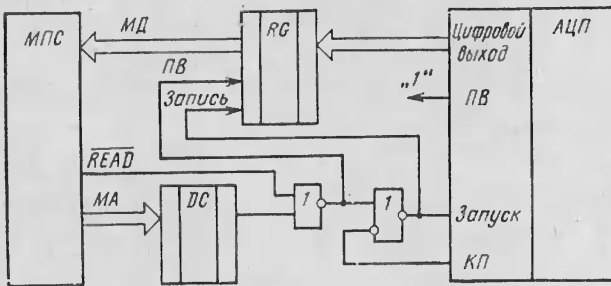


Рис. 8. Сопряжение АЦП с МПС с использованием двойной записи данных

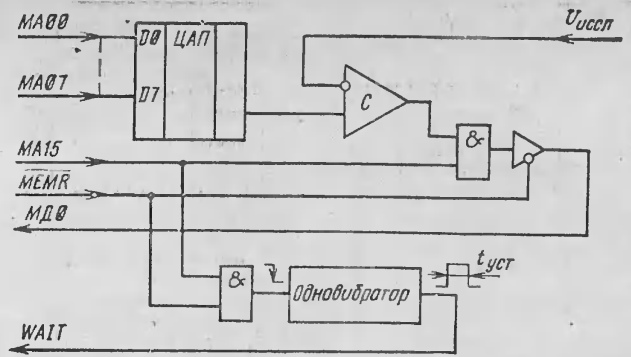


Рис. 9. Функциональная схема АЦП с программным преобразованием информации

АЦП для подобных применений не оправданно из-за большой стоимости и сложности подключения к МПС. Наиболее простым является АЦП последовательного приближения на основе ЦАП, компаратора (рис. 9), программного управления процессом преобразования от МП БИС КР580ИК80 (Программа 1). Для упрощения программы и сокращения числа элементов схемы предусмотрено обращение к ЦАП и компаратору, как к ячейкам памяти. Особенность схемы — подключение ЦАП к младшим восьми разрядам магистрали адреса МПС, что дает возможность одной командой MOVA, M выдавать код на ЦАП и считывать с компаратора результат преобразования (компаратор подключен к магистрали данных через тристабильный буфер). Одновибратор служит для перевода МПС в режим ожидания на время установления выходного сигнала ЦАП. Для увеличения числа обрабатываемых сигналов в схему нужно включить аналоговый мультиметр, устройство выборки-хранения и изменить программу.

ПРОГРАММА 1
 МОНИТОР УМПК-80
 АЦ-НЕСТАНДАРТНАЯ ПРОГРАММНАЯ РЕАЛИЗАЦИЯ
 АНАЛОГО-ЦИФРОВОГО ПРЕОБРАЗОВАНИЯ

0800	210008	ADC LXI	H, 0800H	УСТАНОВКА КАРТЫ
				ПАМЯТИ
0803	010008	LXI	B, 0800H	УСТАНОВКА РЕГИСТРОВ
				ПОСЛЕДОВАТЕЛЬНОГО
				ПРИБЛИЖЕНИЯ
0806	78	MOV	A, B	
0807	81	CNTL	C	
0808	6F	MOV	L, A	
0809	7E	MOV	A, M	СЧИТЫВАНИЕ ЗНАЧЕНИЯ
				С КОМПАРАТОРА
080A	A7	ANA	A	
080B	E21108	JPO	SHF1	ПРОВЕРКА СООТНОШЕНИЯ
080E	78	MOV	A, B	ИССЛЕДУЕМОГО И
080F	81	ORA	C	ТЕСТ-СИГНАЛА
0810	4F	MOV	C, A	
0811	78	SHFT	MOV	A, B
0812	1F	RAR		СДВИГ ТЕСТ-БИТА В
0813	47	MOV	B, A	СТОРОНУ МЛ. РАЗРЯДОВ
0814	D20708	JNC	CNTL	ЕСЛИ ПРЕОБРАЗОВАНИЕ НЕ
				ЗАКОНЧЕНО, ПРОДОЛЖЕНИЕ
				ОЖИДАНИЯ
0817	76	HLT		ОСТАНОВ

Практическое изучение описанных методов сопряжения преобразователей с МПС осуществляется на основе периферийных устройств УМПК-80/МР, входящих в лабораторию для изучения микропроцессорных систем [3]:

устройства для исследования способов обмена данными между МПС и ЦАП с помощью регистров-фиксаторов и канала ПДП, а также изучения различных программных реализаций АЦП, использующих методы поразрядного уравнивания, слежения, последовательного счета (рис. 10);

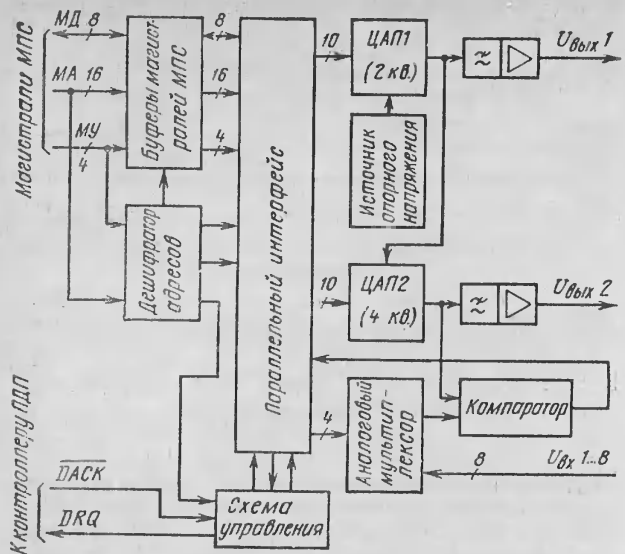


Рис. 10. Структура устройства для изучения методов сопряжения ЦАП и МПС

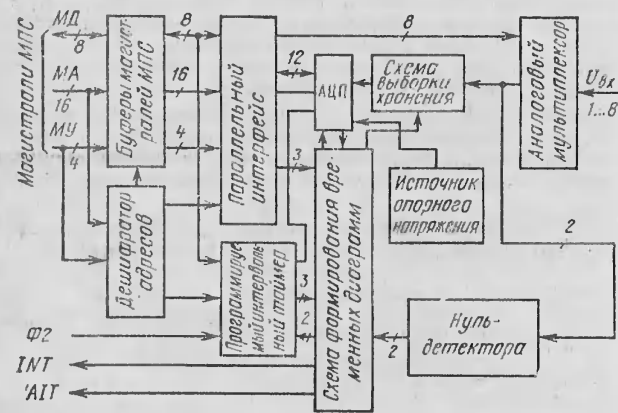


Рис. 11. Структура устройства для изучения методов сопряжения АЦП и МПС

устройства для изучения всех способов сопряжения АЦП с МПС, построенных с использованием счетных методов преобразования (рис. 11). Использование программируемого интервального таймера дает возможность программировать число цифровых разрядов, устанавливаемых в цикле преобразования, а применение аналогового мультиплексора на восемь входов — обрабатывать различные аналоговые сигналы.

Описанные устройства могут быть использованы не только для изучения методов сопряжения МПС, ЦАП и АЦП, но и для реализации на их основе различных информационно-измерительных систем.

Адрес для справок: 103498, Москва, К-498, МИЭТ, телефон: 534-64-96.

ЛИТЕРАТУРА

1. Микроэлектронные цифроаналоговые и аналого-цифровые преобразователи / Б. Г. Федорков и др. — М.: Радио и связь, 1984.
2. Аналоговые и цифровые микросхемы / Под ред. С. В. Якубовского. — М.: Радио и связь, 1985.

3. Преснухин Л. Н., Панфилов Д. И., Романенко О. А., Шаролин С. Г. Микропроцессорная лаборатория по изучению микропроцессорных комплектов с фиксированным набором команд // Микропроцессорные средства и системы. — 1985. — № 1. — С. 77—81.

Статья поступила 8 января 1987 г.

УДК 681.327.21

А. В. Станкевич

КЛАВИАТУРА И БЛОК ИНДИКАЦИИ ДЛЯ ПРИБОРОВ НА БАЗЕ МИКРОПРОЦЕССОРА К1801ВМ1

Применение в составе небольших измерительных приборов стандартных клавиатур и устройств отображения нецелесообразно. Экономичнее разработать малогабаритные специализированные клавиатуры и блоки индикации с минимально возможными аппаратными и программными затратами [1—3].

В отличие от указанных в настоящей работе рассматриваются принципы построения клавиатуры и блока индикации приборов для физического эксперимента, приводятся принципиальные схемы и алгоритмы работы блоков, ориентированные на применение 16-разрядного микропроцессора К1801ВМ1, предлагаются аппаратный способ подавления дребезга контактов клавиш и организация работы клавиатуры по прерыванию. Предложенные принципы позволяют при примерно одинаковых аппаратных затратах сократить объем управляющих программ, исключив программные секции вычисления кода нажатой клавиши и подавления дребезга контактов.

Анализ работы приборов для физического эксперимента позволяет выделить три основных стадии их функционирования:

ввод исходных данных, необходимых для проведения эксперимента (габаритные размеры образцов, зондов, коэффициенты, позволяющие корректировать реальные характеристики датчиков и т. п.);

проведение самого эксперимента; обработку полученного экспериментального материала, выдачу искомым физическим характеристикам исследуемого процесса, перевод прибора в исходное состояние.

В соответствии со стадиями функционирования прибора целесообразен следующий принцип управления

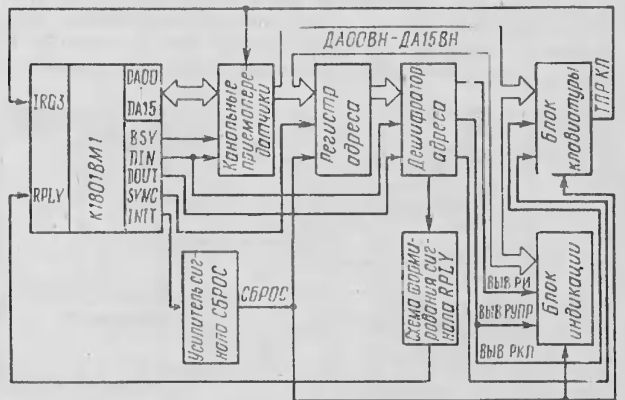


Рис. 1. Схема подключения блоков индикации и клавиатуры к микропроцессору К1801ВМ1

работой клавиатуры и блока индикации. Вся первую и часть третьей стадии, связанную с выдачей искомым физических характеристик либо с выводом на блок индикации точек полученной экспериментальной кривой, микропроцессор будет занят только управлением работой клавиатуры и блока индикации. Проведение эксперимента и обработка экспериментального материала — это выполнение функции клавиши «Пуск». Следовательно, основной режим работы микропроцессора в составе прибора — режим контроллера (устройства управления) клавиатуры и блока индикации (рис. 1). В этом

случае главная управляющая программа обеспечивает сканирование клавиш клавиатуры и динамическую индикацию. Функцию, соответствующую нажатой клавише, может выполнить программа, обслуживающая прерывание от клавиатуры.

Канальные приемопередатчики шины адресов-данных, регистр адреса и дешифратор адреса целесообразно выполнить общими для всех блоков прибора. Сигналы Выв РУПР, Выв РКЛ, Выв РИ вырабатывает дешифратор адреса при обращении в канальном цикле Выв

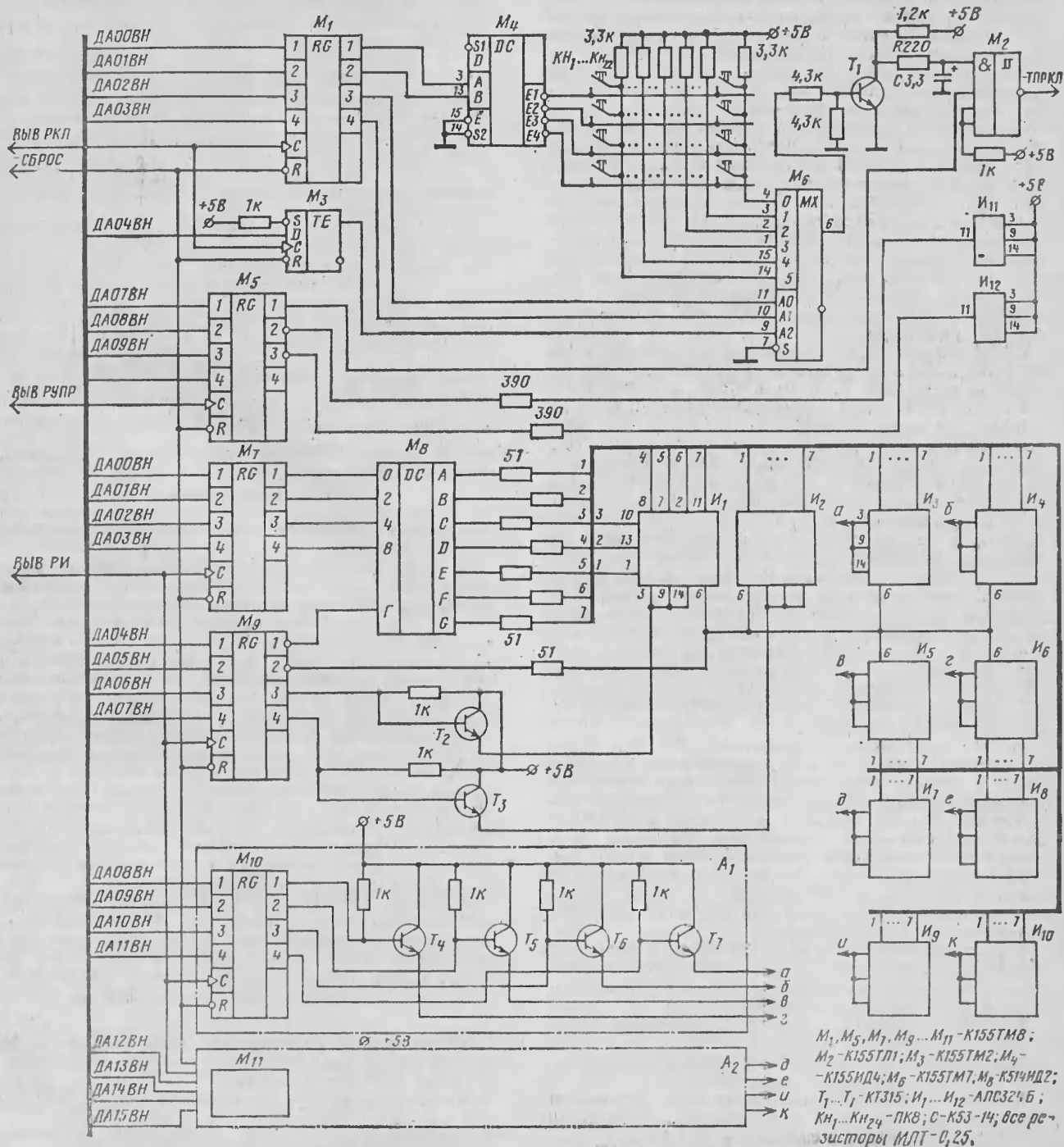


Рис. 2. Принципиальная схема клавиатуры и блока индикации

$M_1, M_5, M_7, M_9 \dots M_{11}$ - К155ТМ8;
 M_2 - К155ТЛ1; M_3 - К155ТМ2; M_4 -
 - К155ИД4; M_6 - К155ТМ7; M_8 - К514ИД2;
 $T_1 \dots T_7$ - КТ315; $I_1 \dots I_{12}$ - АПС324Б;
 $K_1 \dots K_{24}$ - ПК8; C - К53-14; все ре-
 зисторы МЛТ-0,25.

ВОД [4] по адресу регистра управления, регистра клавиатуры и регистра индикации соответственно (рис. 2).

При указанной на схеме разрядности регистра клавиатуры M_1 , M_2 можно организовать клавиатуру, содержащую 32 клавиши. Размер клавиатуры меняется при изменении разрядности регистра клавиатуры (с добавлением или уменьшением числа клавиш).

Микропроцессор сканирует клавиатуру, выводя в регистр клавиатуры двоичные коды от 00000 до 10111. Сигнал на выходе мультиплексора M_6 появится только при нажатии клавиши и при занесении кода, соответствующего этой клавише, в регистр клавиатуры M_1 , M_2 . Дребезг контактов клавиш подавляется с помощью интегрирующий RC-цепочки. Благодаря триггеру Шмитта M_2 установка и снятие сигнала происходят «чисто». Этот сигнал (как прерывающий) подается на вход $IRQ3$ микропроцессора K1801ВМ1 через схему привязки к временной диаграмме процессора [4].

Как показало макетирование клавиатуры, при использовании кнопок ПК-8 для надежного подавления дребезга достаточно иметь постоянную времени RC-цепи 2...3 мс. При номиналах, указанных на схеме, сигнал — ТПР КЛ установится с задержкой порядка 2 мс, а снимется — с задержкой порядка 10 мс. Сигнал — ТПР КЛ используется и для обнаружения момента отпущения нажатой клавиши (чтобы исключить повторное выполнение функции ранее нажатой клавиши). Этот сигнал выводится на каналыные приемопередатчики и считывается в канальном цикле ВВОД при обращении к регистру управления.

Цифровым клавишам целесообразно присвоить коды от 00000 до 01001, представляющие собой двоично-десятичные коды цифр от 0 до 9. Коды от 01010 и далее лучше использовать для функциональных клавиш.

Индикатор имеет 12 разрядов. Из них 6 ($I_1 \dots I_6$) отведены под мантиссу отображаемого числа, два (I_7 , I_8) — под порядок, по одному (I_{11} и I_{12}) — под минус мантиссы и порядка, два разряда (I_9 , I_{10}) — для индикации ошибок (при выполнении арифметических операций, канальных, ввода и т. п.). Шесть разрядов, отведенные под мантиссу, используются для ввода на индикатор содержимого памяти микропроцессора (проверочный режим) в восьмеричном коде. В этом случае занятая, знаки и порядок не индицируются. При необходимости некоторые разряды индикатора могут быть исключены. Индикация минусов мантиссы и порядка — статическая. Их работой управляют два разряда регистра управления M_3 (третий разряд используется для запрещения прерывания клавиатуры). Регистр управления можно совместить с общим регистром управления (или состояний) прибора.

Регистр индикатора — 16-разрядный $M_7 \dots M_{11}$. Четыре младших разряда (0...3) предназначены для занесения в регистр двоично-десятичного кода выводимой цифры, пятый разряд — для индикации запятой, четвертый — для гашения разрядов индикатора, а остальные десять разрядов — для динамической индикации. Для приведенного индикатора необходим буфер в ОЗУ прибора с количеством ячеек, равным числу цифровых разрядов индикатора. В этих ячейках (в нашем случае десяти) хранятся коды выводимой цифры, а также признаки запятой и гашения разряда.

В тексте программы (на рис. 3 приведен ее фрагмент) имя BUF1 соответствует первой ячейке буфера индикатора, RKL — регистр клавиатуры, RI — регистр индикатора, RUPR — регистр управления.

Временная задержка в главной управляющей программе необходима для ожидания срабатывания схемы подавления дребезга. Необходимо, чтобы эта программная задержка была больше аппаратной.

В главной управляющей программе перед фрагментом сканирования клавиатуры и динамической индикации необходимо предусмотреть секцию инициализации (должны быть выполнены установка указателя стека, а также начальная установка всех ячеек и флажков, ис-

```

FRAGMENT ПРОГРАММЫ ДЛЯ СКАНИРОВАНИЯ КЛАВИАТУРЫ
И ДИНАМИЧЕСКОЙ ИНДИКАЦИИ

RUPR=177460
RKL=177474
RI=177470
; KLAU - СТАРТОВЫЙ АДРЕС ПРОГРАММЫ ОБСЛУЖИВАНИЯ ПЕРЕРЫВАНИЯ
; ОТ КЛАВИАТУРЫ

MOU #KLAU, @#270 ; УСТАНОВКА СТАРТОВОГО АДРЕСА
CLR @#272 ; ПЕРЕРЫВАЮЩЕЙ ПРОГРАММЫ
CLR ZBI ; ОБНУЛЕНИЕ УКАЗАТЕЛЯ ЗАПОЛНЕНИЯ
; БУФЕРА ИНДИКАТОРА
BIS #260, @#RUPR ; РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЯ КЛАВИАТУРЫ
CLR MF ; ОБНУЛЕНИЕ ЯЧЕЙКИ ФЛАЖКОВ
5x: MOU #27, R2 ; ЗАНЕСЕНИЕ СЧЕТЧИКА СКАНИРОВАНИЯ
; КЛАВИАТУРЫ (ССК)
7x: MOU R2, @#RKL ; ЗАПИСЬ ССК В РКЛ
MTPS #0 ; РАЗРЕШЕНИЕ ПЕРЕРЫВАНИЯ В СЛОВЕ
3x: CLR R0 ; СОСТОЯНИИ ПРОЦЕССОРА
; ОБНУЛЕНИЕ УКАЗАТЕЛЯ СЧТЫВАНИЯ
; БУФЕРА ИНДИКАТОРА
MOU #100, R1 ; НАЧАЛЬНОЕ ЗНАЧЕНИЕ ДЛЯ
; ДИНАМИЧЕСКОЙ ИНДИКАЦИИ
2x: MOU BUF1(R0), R4 ; ВЫВОД ОЧЕРЕДНОЙ ЦИФРЫ
BIS R1, R4 ; В РАЗРЯД
; ИНДИКАТОРА
1x: MOU R4, @#RI ; ИНДИКАТОРА
MOU #100, R3 ; ВРЕМЕННАЯ ЗАДЕРЖКА
SOB R3, 1x
ASL R1
;
ADD #2, R0
CMP #22, R0 ; ЦИКЛ ИНДИКАЦИИ ОКОНЧЕН?
BGE 2x ; NET
MTPS #260 ; ЗАПРЕТ ПЕРЕРЫВАНИЯ В ССП
BIT #1, MF ; ПЕРЕРЫВАНИЕ БЫЛО?
BEQ 5x ; NET
TST @#RUPR ; КЛАВИША ОПУЩЕНА? (-ТПР КЛ
; ВЫВОДИТСЯ НА ЛИНИЮ ДА 15)
PGE 3x ; NET
BIC #1, MF ; СБРОСИТЬ ФЛАЖОК ПЕРЕРЫВАНИЯ
SOB R2, 7x ; СКАНИРОВАНИЕ КЛАВИАТУРЫ
; ОКОНЧЕНО? - NET
ER 6x ; ДА

```

Рис. 3. Фрагмент главной управляющей программы, обеспечивающий сканирование клавиатуры и динамическую индикацию

пользуемых в программе, обслуживающей прерывание блока клавиатуры).

Не приводим программ, реализующих функции нажатых клавиш, поскольку функциональное назначение клавиш прибора зависит от его применения. Однако общими для прерывающей программы будут следующие действия: эвакуация $R0 \dots R4$; установка «Лог.1» в ячейку флажков (прерывание было) MF; выполнение функции нажатой клавиши; увеличение на два указателя заполнения буфера индикатора (только для цифровых клавиш); возврат в главную управляющую программу.

При выполнении функции клавиши «Пуск» необходимо также сразу после входа в прерывающую программу заблокировать клавиатуру, а перед выходом из программы снова разрешить прерывание клавиатуре. Такая последовательность действий устраняет возможность прерывания эксперимента при случайном нажатии какой-либо клавиши. Если продолжительность эксперимента не превышает несколько сот миллисекунд, то можно не запрещать прерывание клавиатуре.

Описанные клавиатура и блок индикации были разработаны для прибора, измеряющего теплофизические характеристики твердых материалов, однако примененный подход может быть полезен при разработке широкого класса приборов для физического эксперимента.

Телефон для справок: 39-53-23, Минск

ЛИТЕРАТУРА

1. Хэммонд Д. Электроника.—1977.— № 1.— С. 64.
2. Коффрион Дж. Технические средства микропроцессорных систем.— М.: Мир, 1983.— 344 с.
3. Абдуллаев Н. Т., Измайлова Л. З., Тургиев Э. А.— ПСУ.— 1983.— № 7.— С. 26—27.
4. Микросхемы интегральные. Серия K1801. Руководство по применению. ОСТ 11 348.918—83.

Статья поступила 24 апреля 1986 г.

КОНТРОЛЛЕР ДИНАМИЧЕСКОЙ ПАМЯТИ

При построении модулей оперативной памяти большой емкости разработчики часто отдают предпочтение БИС динамических ОЗУ из-за их наибольшей среди существующих микросхем ОЗУ информационной емкости и сравнительно низкой стоимости, несмотря на некоторые недостатки (относительно низкое быстродействие, необходимость проведения регенерации, сложность управления и др.).

Наибольшую трудность при проектировании модулей памяти на микросхемах динамических ОЗУ вызывает необходимость регенерации информации. Это автоматически решается при разработке устройств с последовательным периодическим доступом ко всем ячейкам, например в дисплеях. Однако в накопителях с произвольным доступом обращение к одним ячейкам частое, а к другим — редкое. Отсутствие регенерации приведет к потере информации и к нарушению нормальной работы системы, поэтому при создании таких устройств необходимо позаботиться об обеспечении периодического доступа к ячейкам ОЗУ.

Вторая трудность — мультиплексирование адреса (этот удобный способ сокращения количества выводов микросхем применяется в большинстве микросхем ОЗУ динамического типа).

Мультиплексированием управляют сигналы строба строки (\overline{RAS}), по приходу которого фиксируются младшие разряды адреса, и столбца (\overline{CAS}), им стробируются старшие разряды. По совпадению сигналов \overline{RAS} и \overline{CAS} осуществляется выборка микросхемы. Записью управляет сигнал разрешения записи (\overline{WE}).

Временная диаграмма циклов обращения к ячейке и регенерации приведена на рис. 1. Основные параметры быстрого действия микросхемы — время выборки относительно сигнала строба адреса столбца t_{CAS} , время выборки относительно сигнала выборки адреса строки t_{RAS} и время сохранения сигнала на выходе после снятия сигнала строба адреса столбца t_{OFF} .

При проектировании устройств с использованием БИС динамических ОЗУ следует учитывать, что время выборки зависит от длительности задержки t_{RCD} между установлением сигналов \overline{RAS} и \overline{CAS} . Если время t_{RCD} меньше минимально допустимого, то микросхема не работает, если оно находится в пределах между $t_{RCD\ min}$ и $t_{RCD\ max}$, то время выборки t_{CAS} возрастает на

величину $(t_{RCD\ max} - t_{RCD})$. Когда t_{RCD} больше $t_{RCD\ max}$, нормальная работа микросхемы не нарушается, но время выборки t_{CAS} увеличивается настолько, насколько t_{RCD} превышает $t_{RCD\ max}$ [1—4].

Режим записи, в котором сигнал \overline{WE} устанавливается в активное состояние раньше сигнала \overline{CAS} , называется режимом ранней записи (он показан на временной диаграмме). При этом выход микросхемы во время действия сигнала \overline{CAS} остается в выключенном состоянии. Если же сигнал разрешения записи подать при установленном сигнале \overline{CAS} , то на выходе с задержкой, соответствующей времени выборки, появятся данные, записанные в ячейку при предыдущем обращении к ней. Такой режим называется «считывание — модификация — запись».

Кроме того, в микросхемах K565PU5, K565PU6 возможен страничный режим обращения, а в микросхеме K565PU7 — слоговый режим, позволяющий записывать и считывать четыре последовательных бита с большой скоростью. Эти режимы ускоряют процедуру обращения, но сопряжены с дополнительными аппаратными затратами на их реализацию в контроллерах ОЗУ и в данной статье не рассматриваются.

Регенерацией управляют, как правило, с помощью выборки строки. Ее можно проводить одним сигналом \overline{RAS} без обращения к отдельным ячейкам строки. Период регенерации T_{REF} относится к основным временным параметрам. Хотя T_{REF} непосредственно с быстродействием не связан, следует учитывать, что регенерация может значительно снизить эффективное быстродействие. Эти потери оцениваются по формуле $A = N \cdot t_{CY} / T_{REF} \cdot 100\%$, где N — число регенерируемых строк, t_{CY} — время цикла обращения.

Для регенерации требуются дополнительные логические схемы, генерирующие адреса регенерации и строб-сигналы (рис. 2). Стоимость дополнительных схем для небольших систем памяти может оказаться настолько высокой, что экономически целесообразнее применить статические микросхемы памяти, хотя их информационная емкость меньше, чем у динамических. Чтобы избежать этого, иногда схему управления регенерацией размещают на кристаллах памяти. Например, в микросхеме K565PU7 встроенный счетчик генерирует адреса регенерации. Для управления этим счетчиком используется приобретающий все большую популярность метод «CAS-перед-RAS» [5]. В технических условиях на применение БИС динамических ОЗУ оговорено, что при нормальной работе микросхемы сигнал \overline{RAS} всегда устанавливается раньше, чем сигнал на входе \overline{CAS} . Поэто-

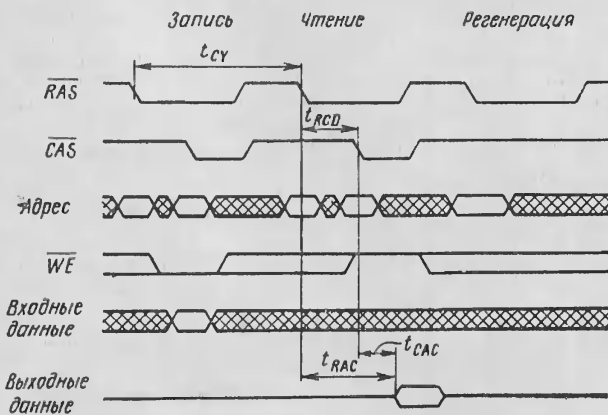


Рис. 1. Временная диаграмма циклов записи, чтения и регенерации микросхем динамического ОЗУ.

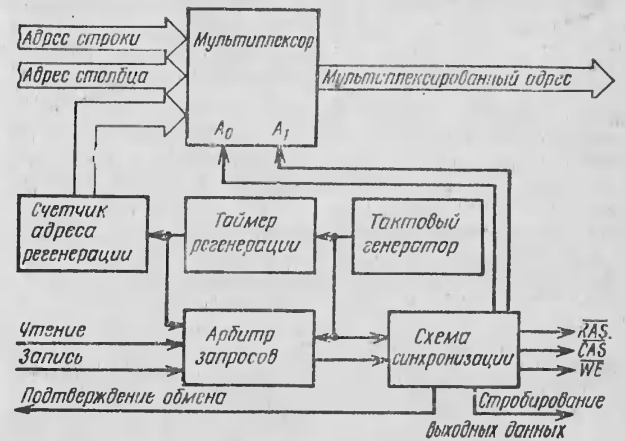


Рис. 2. Структурная схема контроллера БИС динамических ОЗУ (в новых микросхемах памяти счетчик адреса регенерации выполнен в составе кристалла БИС)

му обратную последовательность этих сигналов можно использовать в качестве признака запуска режима регенерации.

Количество циклов регенерации определяется числом строк микросхемы памяти и конфигурацией накопителя. Для БИС ОЗУ емкостью 16К бит и 64К бит необходимо за время не более 2 мс обратиться к 128 строкам. Для ОЗУ емкостью 256К бит требуется провести 512 циклов регенерации не более чем за 8 мс.

Арбитр (схема, управляющая обращениями и регенерацией) — наиболее ответственное устройство в контроллерах динамического ОЗУ. Все сигналы, подаваемые на входы микросхемы памяти, должны соответствовать временной диаграмме ее работы. Эту функцию реализует схема синхронизации, формирующая сигналы управления микросхемами памяти и вспомогательные сигналы. Арбитр разрешает или запрещает выдачу управляющих сигналов на матрицу памяти (рис. 2).

Существует несколько распространенных методов построения арбитра. Для «прозрачного» обращения регенерации используется синхронизация работы арбитра с устройством, осуществляющим обращение [6, 7].

При этом можно избежать циклов ожидания во время операций записи и считывания, так как сигналы управления подаются на БИС ОЗУ синхронно с сигналами записи и чтения на выходах процессора (т. е. скорость обмена информацией максимальна). Регенерация проводится в момент гарантированного отсутствия обращений к ОЗУ, например в тактах выполнения команды процессором. Важно, чтобы во временной диаграмме работы активного устройства были признаки «свободных мест» для циклов регенерации. Недостатки метода — неуниверсальность модулей ОЗУ, а также трудности обеспечения режима прямого доступа или ожидания, реализация которых требует усложнения схемы арбитра.

При построении блоков псевдостатического ОЗУ используется более сложный метод. Он позволяет создать универсальные модули памяти, совместимые со стандартными системными шинами.

Когда таймер покажет, что операции регенерации необходимы, они инициируются арбитром. При обращении процессора в момент проведения цикла записи, чтения или регенерации арбитр переводит его в состоя-

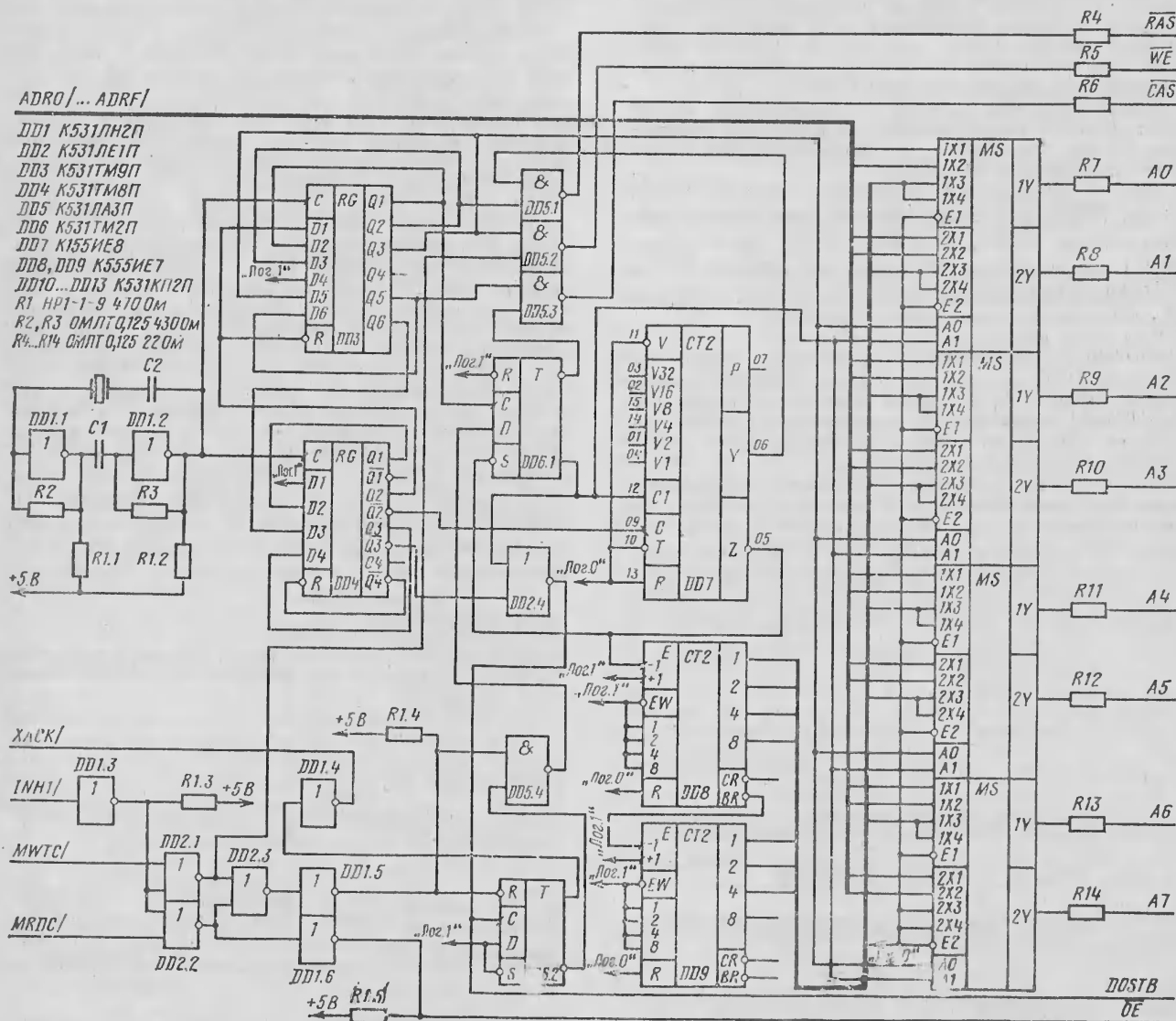


Рис. 3. Принципиальная схема контроллера БИС динамических ОЗУ.

ние ожидания, не давая сигнал подтверждения обмена. Если же запросы на обращение от процессора или на регенерацию от таймера придут одновременно, арбитр сам решает, какой из запросов будет обслужен, а какой поставлен в очередь. В такой схеме модуль памяти универсален: его можно использовать с различными типами процессоров и устройств управления прямым доступом к памяти. Такая схема не нуждается во внешней синхронизации.

Принципиальная схема контроллера, реализующая указанный принцип построения псевдостатических модулей ОЗУ, ориентирована на работу с системной шиной И41 (рис. 3). Она состоит из синхронизатора на основе сдвигового регистра DD3, DD4, арбитра регенерации на микросхемах DD1, DD2, DD5, DD6, таймера регенерации DD7, счетчиков регенерируемой строки DD8, DD9, мультиплексоров адреса DD10... DD13 и выходного буферного регистра данных (на рис. 3 не показан). Устройство рассчитано на работу с микросхемой памяти К565РУ5 или К565РУ6 с любой буквой (период регенерации задается изменением коэффициента пересчета таймера DD7). При увеличении разрядности счетчика регенерации и присоединении мультиплексора еще для одного разряда адреса устройство может управлять СВИС К565РУ7 в режиме «регенерация сигнала RAS». Тактовая последовательность (CLC) для работы контроллера формируется генератором на элементах DD1.1, DD1.2.

Цикл работы схемы (рис. 4) разбит на девять тактов. Число тактов в полном цикле и интервалах RAS, CAS, WE оптимизировано с учетом технических условий на применение микросхем ОЗУ. Первые четыре такта формируют задержку между последовательными сигналами RAS. Кроме того, во втором такте накопленное в таймере регенерации значение увеличивается на единицу и через заданное число циклов на выходе 05 таймера формируется запрос на регенерацию. Этот сигнал сбрасывает триггер регистрации запросов на обращение в состояние «отсутствие обращения». При этом мультиплексоры DD10... DD13, управляемые триггером, переключаются на передачу адреса со счетчиков регенерации на входы микросхем ОЗУ. Высокий уровень, устанавливающийся при наличии запроса на регенерацию на выходе 06 таймера, разрешает выдачу на матрицу памяти сигнала RAS (формируется в тактах с четвертого по восьмой). В нулевом такте цикла управляющие сигналы сбрасываются в неактивное состояние, запрос на регенерацию снимается (это увеличивает на единицу значение адреса регенерируемой строки). За счет применения таймера в описываемой схеме регенерация производится не в каждом «свободном» цикле, а только по запросам таймера. Это сводит к минимуму

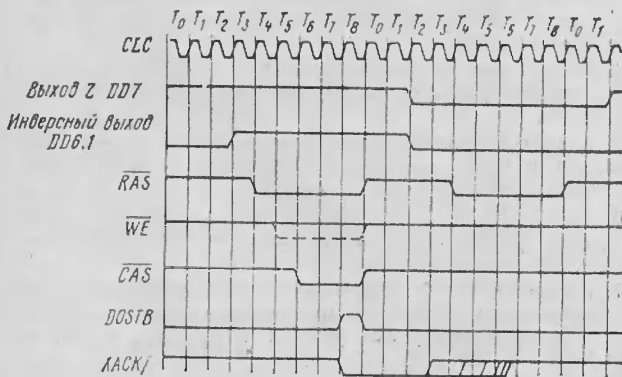


Рис. 4. Временная диаграмма циклов работы контроллера БИС динамических ОЗУ

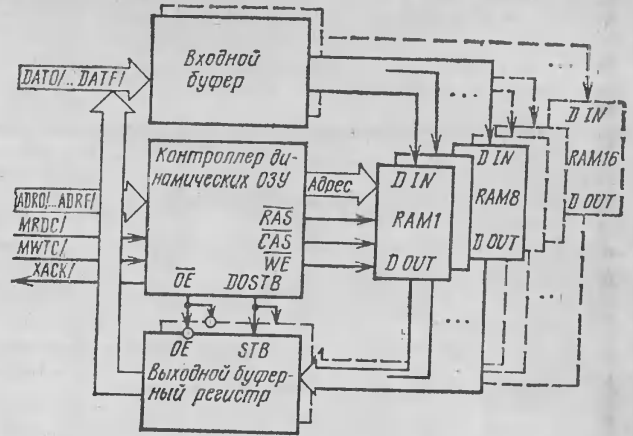


Рис. 5. Структурная схема модуля оперативной памяти

простой процессора из-за ожидания окончания цикла регенерации. Следовательно, эффективное быстродействие микросхем памяти повышается, а рассеиваемая ими мощность снижается.

При отсутствии запроса на регенерацию в третьем такте анализируется, есть ли запрос на обращение. Он формируется элементом DD5.4 по наличию сигнала записи или считывания (при отсутствии сигнала подтверждения обмена XACK/). Если запрос на обращение есть, то триггер DD6.1 переходит в состояние «обращение», разрешая выдачу управляющих сигналов (RAS на четвертом такте и CAS на шестом). В пятом такте переключаются мультиплексоры, передавая в ОЗУ код-адрес столбца. В том же такте при наличии сигнала записи в память MWTC формируется сигнал WE (на один такт раньше, чем сигнал CAS, обеспечивая тем самым режим ранней записи).

Информация, считанная из микросхем, фиксируется в выходном регистре по сигналу DOSTB. Одновременно формируется сигнал подтверждения обмена XACK/. Задержка сигнала DOSTB относительно сигнала CAS определяется временем установления информации t_{CAS} на выходе микросхемы ОЗУ (в данном случае t_{CAS} равно t_{CASmin}). Кроме того, t_{CAS} зависит от эквивалентной емкости на выходе микросхемы памяти. При суммарной емкости не более 20 пФ, т. е. при использовании только одного банка микросхем ОЗУ (рис. 5), сигнал DOSTB формируется, как показано на схеме. При увеличении числа банков емкость на выходе БИС ОЗУ возрастает, поэтому целесообразно использовать в качестве stroba данных сигнал CAS, а триггер, формирующий сигнал «подтверждение обмена» устанавливать переходом CAS в неактивное состояние. В этом случае для выборки банков в схему надо ввести дешифратор, например К531ИД14.

Для максимального быстродействия тактовую частоту f_{CLC} выбирают равной $9/t_{CY}$ (t_{CY} — время полного цикла обращения, приведенное в технических условиях на применение микросхемы).

Если контроллер не должен обеспечивать высокие скорости обращения, целесообразно заменить микросхемы серии К531 сериями К155 или К555 (это снизит энергопотребление модулей памяти и упростит борьбу с помехами по цепям питания).

Схему контроллера для БИС К565РУ7 можно существенно упростить, исключив счетчики регенерации и часть мультиплексоров (рис. 6). Сигнал управления регенерацией с выхода таймера меняет местами сиг-

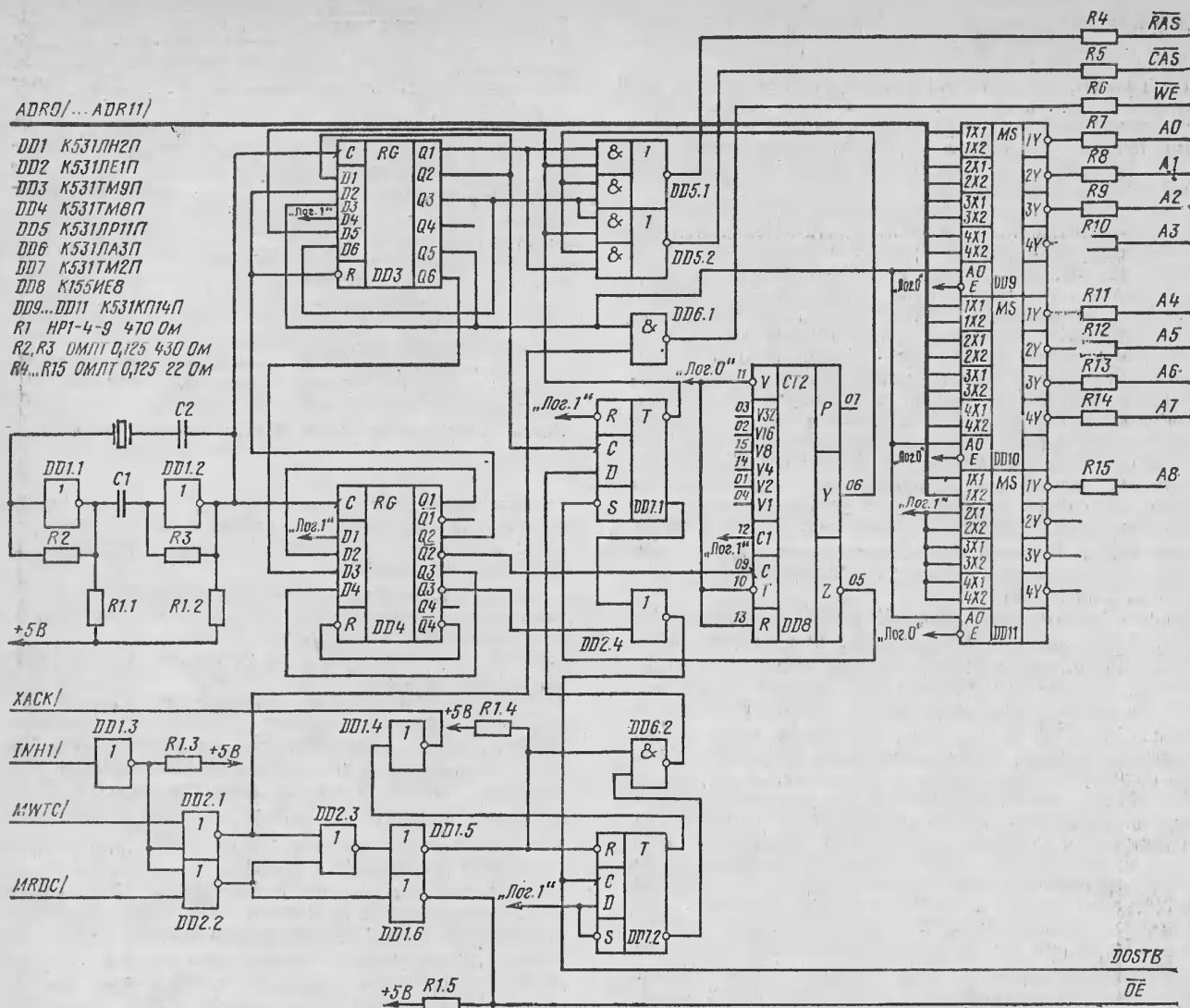


Рис. 6. Принципиальная схема контроллера БИС динамических ОЗУ K565PY6

налы \overline{RAS} и \overline{CAS} , переводя микросхему в режим регенерации очередной строки.

Период регенерации в приведенных схемах задается установкой коэффициента пересчета таймера и определяется по формуле $T_{рег} = 576 N / (f_{CLM})$, где N — число регенерируемых строк; $M = V_{32} \cdot 2^5 + V_{16} 2^4 + \dots + V_1$; $V_1 \dots V_{32}$ — значение уровней на соответствующих входах таймера.

Телефон для справок: 534-22-02, Москва

ЛИТЕРАТУРА

1. Лушников А. С., Однолько А. Б., Соломонович В. И., Широков Г. А. Динамическое ЗУПВ K565PY5 емкостью 64К//Электронная промышленность. — 1983. — № 4. — С. 37.
2. Микросхема динамического ОЗУ K565PY5//Микропроцессорные средства и системы.—1985. — № 3. — С. 91.

3. Беспалов В. П., Ильюшенко А. С., Макаров А. И. и др. Динамическое ЗУПВ KР565PY6 емкостью 16К//Электронная промышленность — 1983. — № 4. — С. 38.
4. Микросхема динамического ОЗУ KР565PY6//Микропроцессорные средства и системы. — 1986. — № 1. — С. 88.
5. Анализ функциональных возможностей, закладываемых в новое поколение динамических ЗУПВ емкостью 256К//Электроника. — 1984. — № 14. — С. 29.
6. Зеленко Г. В., Панов В. В., Попов С. Н. Модуль динамического ОЗУ//Радио. — 1983. — № 10. — С. 28.
7. Еремин Ю. А., Морозов А. Г. Контроллер динамического ОЗУ для микропроцессорных устройств //Микропроцессорные средства и системы. — 1986. — № 3. — С. 75.

Статья поступила 15 февраля 1987 г.

Л. В. Петров, В. П. Ляшев, Т. А. Соловьева

УСТРОЙСТВО СОПРЯЖЕНИЯ НАКОПИТЕЛЯ НА МАГНИТНЫХ ЛЕНТАХ С МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Предлагается простой и эффективный вариант устройства сопряжения НМЛ типа ИЗОТ СМ 5300.01, размещенного на односторонней плате размером 135×240 мм, вставляемой непосредственно в канал микроЭВМ.

Центральный процессор микроЭВМ «Электроника 60» с периодом 2 мс осуществляет цикл регенерации динамических микросхем памяти в течение 150 мс, во время которого выполнение текущей программы прекраща-

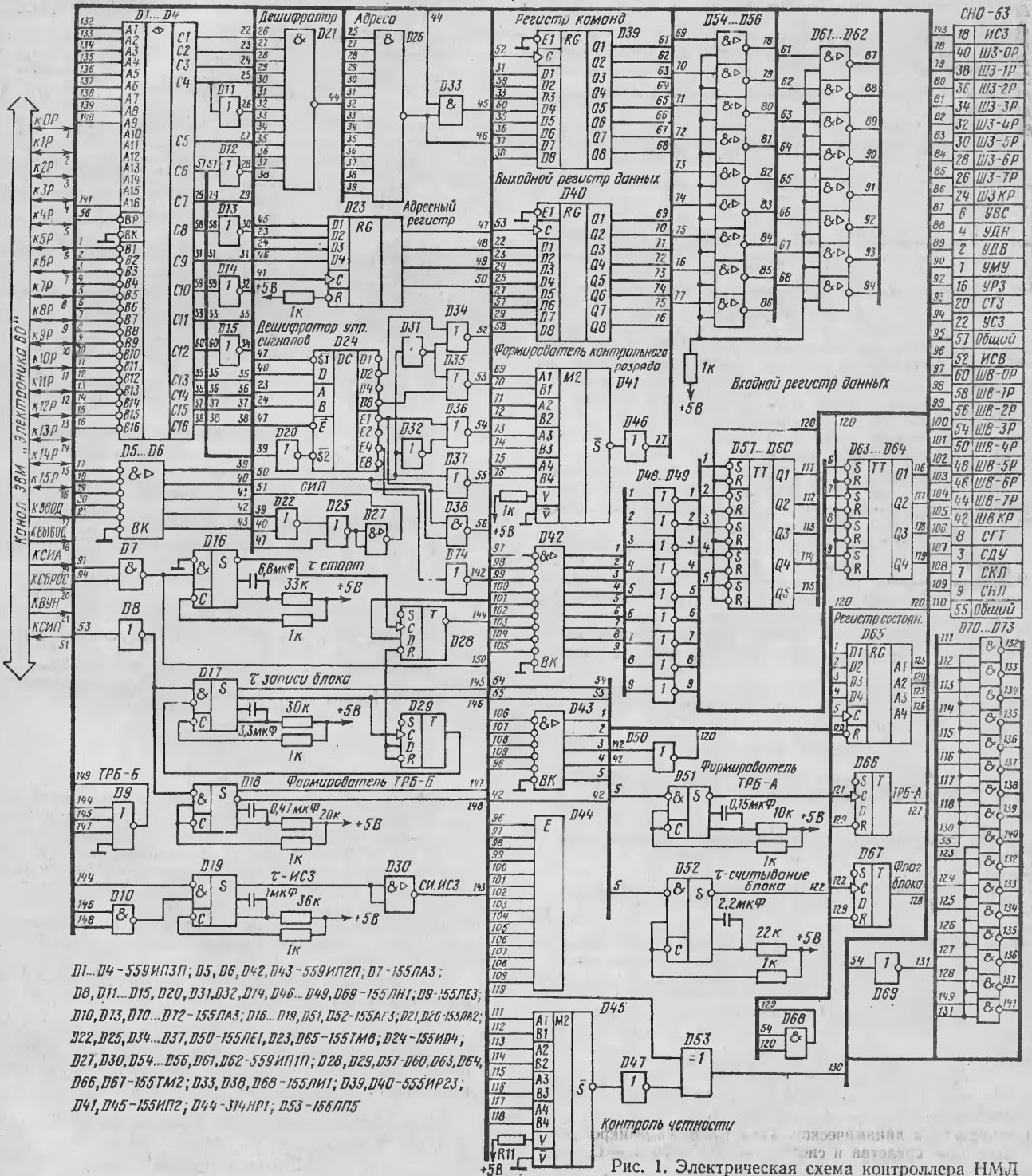


Рис. 1. Электрическая схема контроллера НМЛ

МОДУЛЬ ЭЛЕКТРИЧЕСКИ ПЕРЕПРОГРАММИРУЕМОГО ПЗУ ДЛЯ МИКРОЭВМ СЕМЕЙСТВА «ЭЛЕКТРОНИКА 60»

Постоянные запоминающие устройства (ПЗУ) [1] служат для хранения программ и данных, постоянно используемых в микроЭВМ. Окончательный вариант этих программ формируется в результате их неоднократной проверки. При таком способе отладки программ каждый раз требуются ПЗУ, содержащие новую информацию. ПЗУ с электрически пережигаемыми перемычками в этом случае неэффективны (их можно использовать только один раз).

Эффективно применить ПЗУ с однократной сменой информации [2]. Информация в таких ПЗУ стирается электрическим способом или с помощью ультрафиолетового излучения (УФ). Предпочтительнее электрически репрограммируемые ПЗУ (РПЗУ), так как информация в них стирается быстрее, чем в РПЗУ с УФ-стиранием, и не требует такого оборудования, как источник УФ-излучения [2]. Применять РПЗУ целесообразно в микроЭВМ, у которых информация

в ПЗУ меняется редко. В этом случае интерфейс РПЗУ можно дополнить аппаратными средствами для репрограммирования.

Наш модуль электрически репрограммируемого ПЗУ (ППЗУ) (16К байт), выполненный на БИС РПЗУ КМ558РРЗ, предназначен для работы в микроЭВМ типа «Электроника 60», созданной на основе КМДП БИС серии К588 [3, 4]. При включении питания элементы интерфейса модуля устанавливаются в режим, позволяющий считывать информацию, хранящуюся в БИС РПЗУ. Для снижения потребляемой мощности можно программно отключать напряжение питания от БИС РПЗУ (после перезаписи содержащейся в них информации в ОЗУ на основе КМДП БИС). В модуле есть аппаратные средства для записи и стирания информации.

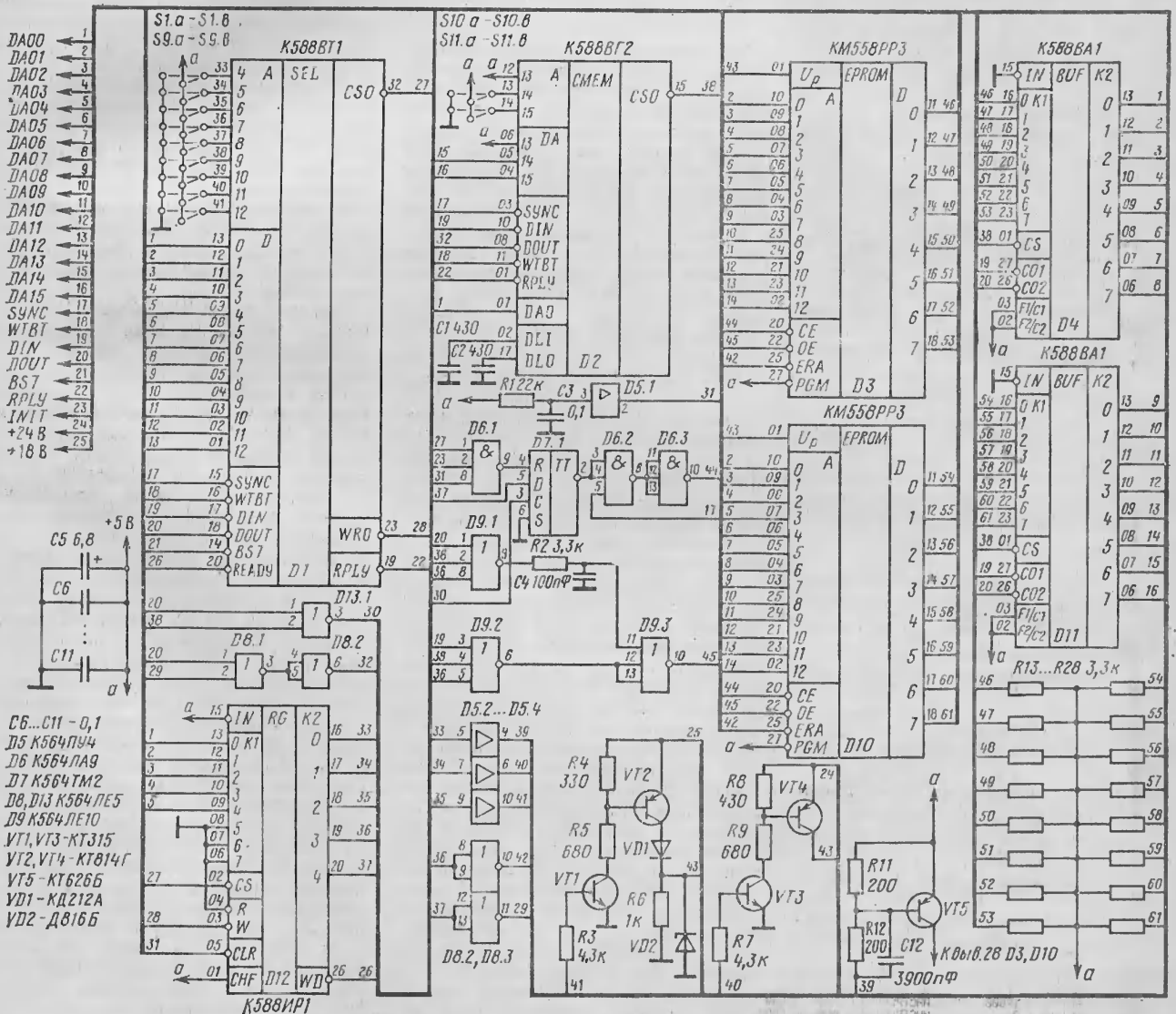


Рис. 1. Принципиальная схема модуля ППЗУ

Режимы работы модуля задаются с помощью регистра режима (PP) D12 K588IP1 (рис. 1), высокий уровень сигналов на выходной шине которого — активный. Регистр K588IP1 используется в неинвертирующем включении. Назначение разрядов его выходной шины следующее: 0 — отключение питания БИС РПЗУ, 1 — подключение источника программирования U_p (+24 В), 2 — подключение источника стирания U_c (+18 В), 3 — разрешение стирания, 4 — разрешение записи. Чтение информации из модуля аналогично циклу микроЭВМ типа «Электроника 60» (рис. 2). Минимальная длительность сигнала разрешения чтения БИС KM558PP3 — 170 нс. Временная диаграмма в режимах программирования и стирания формируется аппаратно-программным способом при подключении источников U_p и U_c (токи записи и стирания 300 мА). Такой способ формирования временных диаграмм обусловлен тем, что длительность цикла записи одного слова в БИС KM558PP3 — 5 мс (рис. 3), а информация стирается за 20 с (рис. 4) по всем адресам одновременно. Допустимые отклонения длительностей указанных циклов $\pm 10\%$.

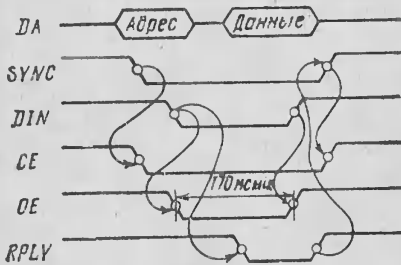


Рис. 2. Временная диаграмма чтения информации в модуле ППЗУ

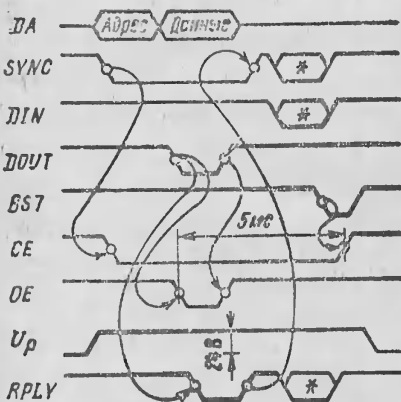


Рис. 3. Временная диаграмма записи информации в модуле ППЗУ (* — уровень сигнала не имеет значения)

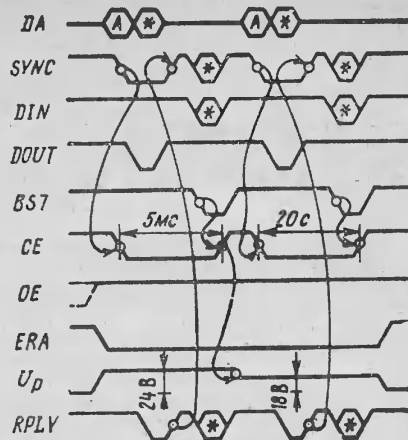


Рис. 4. Временная диаграмма стирания информации в модуле ППЗУ (* — то же, что на рис. 3)

При включении источника питания формирователь сигнала начальной установки (элементы D5.1, R1, C3) вырабатывает импульс, сбрасывающий триггер формирования циклов записи и стирания (ТФЦЗС) D7.1 K564TM2, и очищает PP. Аналогично воздействует на ТФЦЗС и PP сигнал сброса INIT, вырабатываемый в микроЭВМ при выполнении команды сброса RESET или при выполнении начального пуска центрального процессора (ЦП) микроЭВМ. При выводе ЦП числа

(X XXX XXX XXX X11 111)₂
по адресу PP, определяемому как
(111 A12 A11 A10 A9 A8 A7 A6
A5 A4 1110)₂,

на выходе CS0 БИС селектора адреса (CA) K588BT1 D1 вырабатывается сигнал, устанавливающий ТФЦЗС в исходное состояние. В PP записывается число (00000)₂. Значения разрядов (X) выводимого числа являются безразличными. Выходная шина PP поддерживается в активном состоянии, поскольку на входе чтения R постоянно присутствует низкий уровень, а информация записывается сигналом записи WRO CA (при наличии сигнала на выходе CS0). Значения разрядов A4...A12, задаваемые на шине адреса (A) CA, должны быть инверсными по отношению к соответствующим разрядам кода адреса, передаваемого по шине данных-адреса (DA), а разряды чисел, выводимых в PP, — инверсными по отношению к разрядам требуемых сигналов управления. Это объясняется тем, что в микроЭВМ типа «Электроника 60» сигналы шины DA — инверсные по отношению к передаваемым по ней числам. CA функционирует под воздействием сигналов управления шиной DA: обмена (SYNC), признака, вывода байтовой информации (WTBT), признака кода адреса внешнего устройства (BS7),

ввода данных (DIN), вывода данных (DOUT), ответа (RPLY) [5].

При нулевом уровне сигнала разряда 0 выходной шины PP открывается ключ на транзисторе VT5, и напряжение питания подается на БИС РПЗУ D3 и D10. Такое состояние элементов модуля ППЗУ — режим чтения информации (под управлением БИС контроллера запоминающего устройства (КЗУ) K588BG2 D2 [6] в течение цикла DIN (см. рис. 1) ЦП микроЭВМ). По отрицательному фронту сигнала SYNC (поступающего через элементы D6.2 и D6.3 на вход выборки кристалла CE D3 и D10) передаваемый по шине DA код адреса записывается во внутренние регистры БИС РПЗУ. Если этот адрес принадлежит адресному пространству модуля ППЗУ, то на выходе выборки кристалла CS0 БИС КЗУ вырабатывается сигнал низкого уровня, стробирующий сигнал DIN. Этот сигнал проходит через элементы D9.2 и D9.3 на вход разрешения выдачи данных OE БИС РПЗУ. Данные из адресованной ячейки БИС РПЗУ считываются и поступают на шину DA микроЭВМ через магистральные приемопередатчики (МПП) D4 и D11 K588BA1. МПП и резисторы R13...R28 согласовывают TTL-уровни сигналов БИС РПЗУ с уровнями сигналов шины DA микроЭВМ на основе КМДП БИС [4]. По сигналу ответа RPLY (формируется БИС КЗУ) заканчивается цикл обмена SYNC.

При записи в PP числа (10010)₂ модуль ППЗУ переходит в режим программирования (см. рис. 3). Открывается ключ на транзисторах VT3 и VT4, и напряжение (+24 В) подается на вход программирования БИС РПЗУ. Адрес записывается во внутренние регистры БИС РПЗУ, как в режиме чтения. По сигналу DOUT, стробируемому сигналом CS0 КЗУ на элементе D13.1, устанавливается ТФЦЗС и на входе CE БИС РПЗУ «зашелкивается» сигнал низкого уровня. Вход OE управляется сигналом DOUT, а прохождение сигналов DIN в следующих циклах обмена ЦП блокируется Цепочкой R2, C4 и элементами D9.1 и D9.3 сигнал DOUT задерживается на время прохождения данных через МПП на шину данных D БИС РПЗУ. После программного формирования цикла записи (5 мс) в PP записывается число (00000)₂. При этом отключается источник U_p , а сигналom CS0 CA сбрасывается ТФЦЗС (снимается сигнал CE БИС РПЗУ). Цикл программирования одного слова заканчивается. Такое состояние элементов модуля ППЗУ соответствует режиму чтения информации. Для запрета формирования сигнала RPLY при нулевом значении 4-го разряда PP служат элементы D8.1 и D8.2 (стробируют сигнал DOUT, поступающий на соответствующий вход КЗУ).

Информация в БИС РПЗУ стирается в два этапа (см. рис. 4). В РР записывается код $(11010)_2$ и ко входу U_p БИС РПЗУ подключается источник U_n . На вход ERA подается сигнал низкого уровня, разрешающий стирание. При обращении в цикле DOUT к ячейке с произвольным адресом из области адресов модуля ППЗУ в ТФЦС «защелкивается» сигнал CE (программно поддерживается в течение 5 мс, как в цикле записи). Затем при выводе в РР числа $(11100)_2$ снимается сигнал CE и напряжение (+18 В) с помощью ключа на транзисторах VT1 и VT2 подается на вход БИС РПЗУ. Интервал стирания (20 с) формируется, как интервал 5 мс. При записи в РР кода $(00000)_2$ стирание заканчивается, и модуль ППЗУ переводится в режим чтения информации. В течение цикла стирания на входе OE БИС РПЗУ поддерживается высокий уровень. Это обеспечивается стробированием сигналов DIN и DOUT сигналом разрешения стирания на элементах D9.1 и D9.2.

Для отключения напряжения питания от БИС РПЗУ в РР необходимо записать число $(00001)_2$. Максимальная потребляемая мощность БИС KM558PP3 в режиме хранения информации 110 мВт, а в режиме обращения — 420 мВт. Мощность, потребляемая модулем ППЗУ (определяется мощностью потребления БИС РПЗУ), в режиме хранения — 220 мВт, а в режиме обращения — 840 мВт. Мощность, потребляемая модулем ОЗУ на основе КМДП БИС (16К байт) [6], в режиме хранения — 20 мВт, а в режиме обращения — 260 мВт. Если информация из модуля ППЗУ переписывается в такой модуль ОЗУ, а затем отключается питание от БИС РПЗУ, то выигрыш в потребляемой мощности в режиме хранения — 11 раз, а в режиме обращения — 3,2 раза.

Интервалы времени циклов записи и стирания в модуле ППЗУ формируются программно программными счетчиками времени [2]. Для этих целей разработаны подпрограммы на языке MACRO-11 [7]. Они функционируют под управлением ОС РАФОС.

В начале выполнения алгоритма записи массива информации, хранящегося в ОЗУ микроЭВМ (рис. 5, 6), устанавливаются начальные адреса ОЗУ и ПЗУ, а также размер записываемого массива. В данном алгоритме размер массива — 256 16-разрядных слов — один блок информации на гибком магнитном диске (ГМД). Начальный адрес ОЗУ заносится в регистр R1 ЦП, начальный адрес программируемой области модуля ППЗУ — в R2, а размер массива записывается в регистр R3 (служит счетчиком слов). Регистр R4 — программный счетчик времени. Перед записью каждого последующего слова в регистр заносится число из

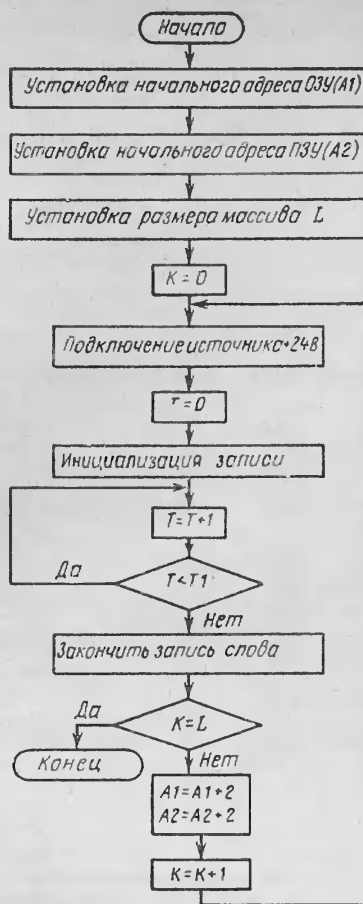


Рис. 5. Блок-схема алгоритма записи информации в модуле ППЗУ

ячейки T1, определяющее длительность временного интервала цикла записи одного слова. Затем в РР выводится число $(10010)_2$, при котором разрешается запись информации, и ко входу U_p БИС РПЗУ подключается источник U_n . Запись инициализируется при выводе информации из ячейки ОЗУ в соответствующую ячейку РПЗУ. С помощью счетчика R4 формируется временной интервал (5 мс), и после вывода в РР кода $(00000)_2$ запись слова завершается, источник U_n отключается. Если массив данных записан в РПЗУ неполностью, то адреса ОЗУ и ПЗУ приращиваются (фактически одновременно с инициализацией записи, так как в программе используется автоинкрементный метод адресации) и цикл записи одного слова повторяется.

Для стирания информации в модуле ППЗУ (рис. 7, 8) при выводе в РР кода $(11010)_2$ ко входу U_p БИС РПЗУ подключается источник U_n и ко вход ERA подается сигнал низкого уровня. Регистры R1 и R2 используются в качестве программных счетчиков времени. В регистр R1 заносится число из ячейки T1, определяющее длительность первого этапа стирания. Затем при выводе числа $(0)_2$ по любому из адресов адресного пространства ячеек модуля ППЗУ инициализируется стирание и с помощью счетчика R1 формируется временной интервал (5 мс). После вывода в РР кода $(11100)_2$ первый этап стирания завершается и ко входу БИС РПЗУ U_p подключается источник U_n . Второй этап стирания реализуется аналогично первому, но

АСЕМБЛЕР MACRO-11/РАФОС

```

;
;
;
; *****
; *
; * WRITE ПОДПРОГРАММА ПЕРЕЗАПИСИ МАССИВА ДАННЫХ ДЛИНОЙ 256 СЛОВ
; * ИЗ ОЗУ В МОДУЛЬ ППЗУ.
; *
; * A1 - НАЧАЛЬНЫЙ АДРЕС ОЗУ;
; * A2 - ЯЧЕЙКА, В КОТОРОЙ ХРАНИТСЯ НАЧАЛЬНЫЙ АДРЕС РПЗУ;
; * CRG АДРЕС РЕГИСТРА РЕЖИМА;
; * T1 НАЧАЛЬНОЕ ЗНАЧЕНИЕ ПРОГРАММНОГО СЧЕТЧИКА ВРЕМЕНИ.
; *
; *****
;
; CRG=166776 ;ЗАДАНИЕ АДРЕСА РЕГИСТРА РЕЖИМА
;
; WRITE: MOV %A1,R1 ;НАЧАЛЬНЫЙ АДРЕС ОЗУ В R1
; MOV A2,R2 ;НАЧАЛЬНЫЙ АДРЕС РПЗУ В R2
; MOV #256,R3 ;РАЗМЕР МАССИВА ДАННЫХ В R3
; WORD: MOV T1,R4
; MOV #0B 01101,0#CRG ;РАЗРЕШЕНИЕ ЗАПИСИ И ПОДКЛЮЧЕНИЕ ИСТОЧНИКА
; ; +24В К ВХОДУ ПРОГРАММИРОВАНИЯ БИС РПЗУ
; TIME: MOV (R1)+,(R2)+ ;ИНИЦИАЛИЗАЦИЯ ЗАПИСИ СЛОВА
; NOP ;ОБРАЗОВАНИЕ ВРЕМЕННОЙ ДИАГРАММЫ
; SOB R4,TIME ;РАБОТЫ БИС РПЗУ
; MOV #0B 11111,0#CRG ;ЗАКОНЧИТЬ ЗАПИСЬ СЛОВА
; SOB R3,WORD ;ПРИСТУПИТЬ К ЗАПИСИ СЛЕДУЮЩЕГО СЛОВА
; RTS PC
;
;
; T1: .WORD 1120
; A1: .BLKW 256.
; A2: .BLKW 1
  
```

Рис. 6. Подпрограмма записи информации в модуле ППЗУ

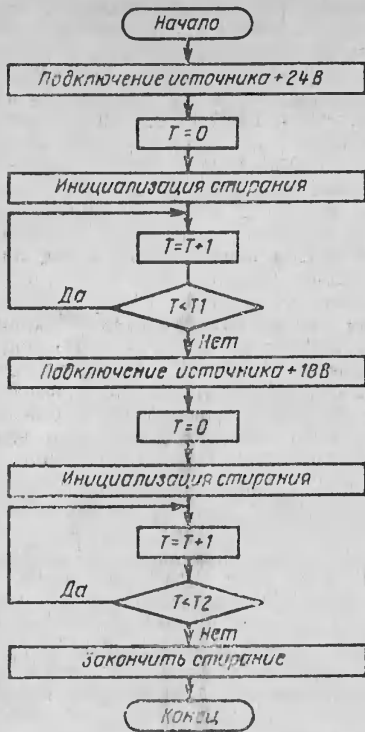


Рис. 7. Блок-схема алгоритма программного формирования цикла стирания информации в модуле ППЗУ.

в связи с большой длительностью временного интервала применяются два вложенных счетчика с использованием регистров R1 и R2. При выводе в PP числа (00000)₂ стирание завершается.

На основе рассмотренных алгоритмов разработана программа (на языке MACRO-11) программирования и стирания информации в БИС РПЗУ, а также верификации (рис. 9). Данные для программирования БИС РПЗУ должны быть подготовлены в отдельном файле на ГМД. При работе с программой в интерактивном режиме выбирается режим работы с модулем ППЗУ и вводится имя файла, в котором находятся данные для программирования или верификации, вводится адрес ячейки, начиная с которого будет располагаться программа. Затем блок информации с ГМД считывается в ОЗУ и (в зависимости от режима) записывается или по слову сравнивается с соответствующей областью адресного пространства модуля ППЗУ. По окончании файла происходит переход на начало программы. В случае

```

;
;*****
;
; CLEAR - ПОДПРОГРАММА СТИРАНИЯ ИНФОРМАЦИИ В МОДУЛЕ ППЗУ.
;
; CRG - АДРЕС РЕГИСТРА РЕЖИМА;
; A2 - ЯЧЕЙКА, В КОТОРОЙ ХРАНИТСЯ НАЧАЛЬНЫЙ АДРЕС РПЗУ;
; T1 - НАЧАЛЬНОЕ ЗНАЧЕНИЕ 1-ГО ПРОГРАММНОГО СЧЕТЧИКА
; ВРЕМЕНИ;
; T2 - НАЧАЛЬНОЕ ЗНАЧЕНИЕ 2-ГО ПРОГРАММНОГО СЧЕТЧИКА
; ВРЕМЕНИ;
; DT - НАЧАЛЬНОЕ ЗНАЧЕНИЕ ВСПОМОГАТЕЛЬНОГО СЧЕТЧИКА.
;*****
;
; CRG=166776 ;ЗАДАНИЕ АДРЕСА РЕГИСТРА РЕЖИМА
;
; CLEAR: MOV #^В 00101,0#CRG ;РАЗРЕШЕНИЕ СТИРАНИЯ И ПОДКЛЮЧЕНИЕ ИСТОЧНИКА
; ; +24В К ВХОДУ ПРОГРАММИРОВАНИЯ БИС РПЗУ
;
; MOV T1,R1
; MOV #0,0A2 ;ИНИЦИАЛИЗАЦИЯ СТИРАНИЯ
TIME1: NOP ;ФОРМИРОВАНИЕ ИНТЕРВАЛА 5 МС
; SOB R1,TIME1
; MOV T2,R2
; MOV #^В 00011,0#CRG ;ПОДКЛЮЧЕНИЕ ИСТОЧНИКА +16В К ВХОДУ
; ; ПРОГРАММИРОВАНИЯ БИС РПЗУ
; MOV #0,0A2 ;ИНИЦИАЛИЗАЦИЯ СТИРАНИЯ
TIME2: MOV DT,R1 ;ФОРМИРОВАНИЕ ИНТЕРВАЛА 20 С
; TIME3: NOP
; SOB R1,TIME3
; SOB R2,TIME2
; MOV #^В 11111,0#CRG ;ЗАКОНЧИТЬ СТИРАНИЕ
; RTS PC
;
; T1: .WORD 1120
; T2: .WORD 11000
; DT: .WORD 1000
; A2: .BLKW 1
;

```

Рис. 8. Подпрограмма стирания информации в модуле ППЗУ

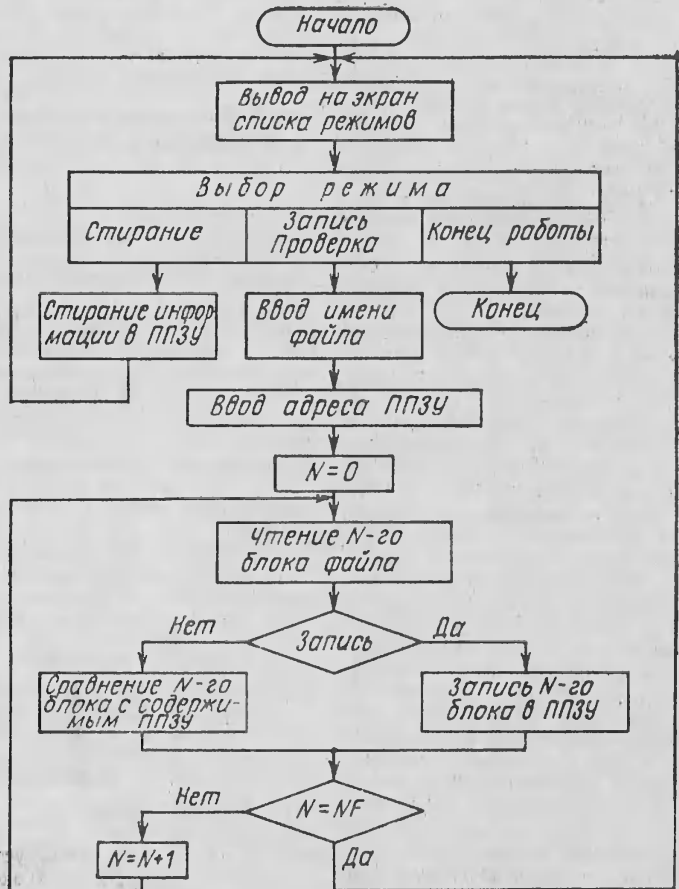


Рис. 9. Блок-схема алгоритма репрограммирования информации в модуле ППЗУ

несовпадения содержимого ячеек БИС РПЗУ и данных на ГМД при верификации на экран дисплея выводятся адрес и содержимое запрограммированной ячейки и соответствующего слова на диске.

Аппаратные и программные средства модуля, позволяющие репрограммировать БИС РПЗУ, дают возможность модулю служить в качестве программатора БИС КМ558РРЗ. Модуль ППЗУ используется в микроЭВМ управления адаптивной антенной решеткой [4]. Он эффективен и в качестве технологической памяти при отладке ПО управляющих микроЭВМ.

Адрес для справок: 103498, Москва, К-498, МИЭТ. Тел. 534-09-08.

ЛИТЕРАТУРА

1. Полупроводниковые запоминающие устройства и их применение / Под ред. А. Ю. Гордонова.— М.: Радио и связь, 1981.
2. Косарев Ю. А., Виноградов С. В. Электрически изменяемые ПЗУ.— Л.: Энергоатомиздат, 1985.
3. Бобков В. А., Шиллер В. А. Комплект КМДП БИС для специализированных 16-разрядных микровычислительных систем с унифицированным интерфейсом // Электронная промышленность.— 1981.— № 4.— С. 32—35.
4. Джиган В. И., Лужецкий С. Г. МикроЭВМ управления фазированной антенной решеткой.— В кн.: Архитектура, схемотехника и математическое обеспечение микропроцессорных систем. Межвуз. сб. /— М.: МИЭТ, 1986, С. 131—136.
5. Никитюк Н. М. Микропроцессоры и микроЭВМ: применение в приборостроении и научных исследованиях.— М.: Энергоатомиздат, 1981.
6. Коннонов Е. В., Тихомиров С. Н., Черняковский Д. Н. Модуль ОЗУ с унифицированным интерфейсом на основе БИС серии КР588 // Электронная промышленность.— 1983.— № 9.— С. 22—23.
7. Экхауз Р., Морис Л. Мини-ЭВМ: Организация и программирование / Пер. с англ. под ред. Г. П. Васильева.— М.: Финансы и статистика, 1983.

Статья поступила 25 ноября 1986 г.

УДК 681.322.1+681.325.5+681.3.022

Ф. М. Яблонский, У. Ю. Усмонов, С. Влахов, И. Косев, Д. Павлов, Г. Радонов

МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКС ДЛЯ ИЗУЧЕНИЯ СРЕДСТВ ОТОБРАЖЕНИЯ ИНФОРМАЦИИ

Учебный комплекс микропроцессорных средств используется на кафедре «Промышленная электроника» Московского энергетического института при изучении дисциплины «Средства отображения информации». Лабораторные работы на основе комплекса позволяют овладеть методами формирования и оценки информационных моделей, навыками разработки отдельных модулей и системы отображения в целом.

В процессе работы можно задавать основные характеристики системы и режимы управления индикаторами, способы сопряжения индикаторов с источниками данных.

Ядро комплекса — серийное учебно-отладочное устройство (УОУ) «Электроника 580» [1]. Знакосинтезирующие индикаторы и дополнительные элементы комплекса (клавиатура, времязадающий блок) подсоединяются к системной шине УОУ, содержащей буферизированные ША и ШД линии системных сигналов. МикроЭВМ УОУ используется как источник информации и как универсальный блок управления отображением.

Применение знаковсинтезирующих индикаторов и модулей на их основе позволяет реализовать и проиллюстрировать цифровые методы управления индикаторами, настроенными на различных физических принципах.

Комплекс можно использовать и для изучения дисплеев на ЭЛТ с применением контроллера ЭЛТ [2, 3].

Определение параметров и характеристик знаковсинтезирующих индикаторов (рис. 1). С помощью программируемого источника напряжения формируется импульс возбуждения индикатора (по частоте, длительности и амплитуде). Блок выборки координат этого воздействия передает на изучаемый элемент индикаторного поля. Свечение, возникающее в любой точке индикаторного поля, через широкоугольную оптику поступает на фотоприемник, а затем через схему выборки и хранения и аналого-цифровой преобразователь вводится в оцифрованном виде в УОУ. Программно может задаваться режим измерения мгновенной или кажущейся силы света.

Установка позволяет снимать электрические параметры и характеристики газоразрядных индикаторных панелей, яркостные и модуляционные характеристики любых типов знаковсинтезирующих индикаторов; определять среднее значение и разброс напряжений возникновения разряда, яркость свечения и разброс по индикаторному полю; воспроизводить кривые восстановления электрической прочности нарастания и спада свечения; определять сеточные пус-

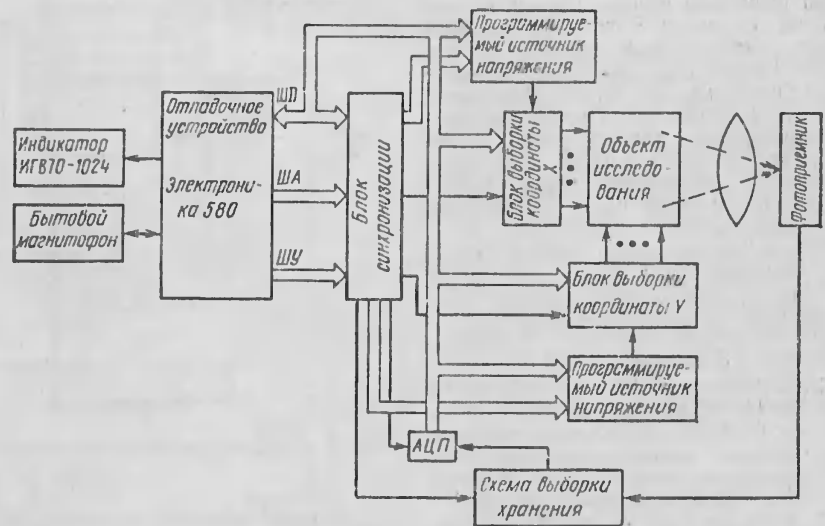


Рис. 1. Структурная схема установки для определения параметров и характеристик знаковсинтезирующих индикаторов

ковые характеристики индикаторных тиратронных матриц со статистической обработкой или без нее.

Отображение цифровой и шкальной информации (рис. 2). Вместо контроллера клавиатуры и дисплея КР580ВВ79 использован программируемый параллельный адаптер КР580ВВ55, что позволяет реализовать все виды динамического управления индикаторами (фазоимпульсный, с разверткой по знакоместам и с разверткой по сегментам). Эти же адаптеры были применены для связи УОУ с другими индикаторами. Установка дает возможность разрабатывать динамические и статические схемы управления цифровыми индикаторами, схемы управления шкальными индикаторами, отображать изменяющиеся во времени показания, формировать уставки для предельных значений в виде рисок и т. д.

Отображение символьной и псевдографической информации (рис. 3). Состав установки: индикаторный модуль ПИУ-1 на газоразрядной индикаторной панели с самосканированием ГИПС16 (в модуль встроено ОЗУ на сдвиговых регистрах на 16 байт), индикаторный модуль ИГВ70-1024/5×7 и символьная клавиатура 15ВВВ-97-006.

Связь индикаторных модулей с УОУ осуществляется через простые порты вывода. На интерфейсной плате расположены времязадающий блок с таймером КР580ВВ153, синхронизированный тактовой частотой микроЭВМ УОУ (2 МГц). Запросы таймера и клавиатуры считываются микроЭВМ с порта ввода.

Кроме того, комплекс позволяет исследовать ряд психофизиологических характеристик операторов, например способность к распознаванию образов на экране. Для этого выводятся фрагмент фигуры и множество вариантов его дополнения до целого. Оператор с помощью клавиатуры вводит свой вариант ответа. МикроЭВМ УОУ фиксирует на экране нарастающее число правильных ответов и ошибок. Программа, поддерживающая данный режим, требует до 4К байта памяти. ОЗУ отладочного устройства имеет только 1К байт, поэтому часть подпрограмм записана в ПЗУ.

Адрес для справок: 105835, Москва, Красноказарменная ул., д. 14, МЭИ, кафедра промышленной электроники. Телефон: 362-78-45.

ЛИТЕРАТУРА

1. Широков Ю. Ф., Осипов Е. И., Жданов В. И., Гутвец Н. И. Учебно-отладочное устройство «Электроника-580» // Микропроцессорные средства и системы.— 1985.— № 2.— С. 80—85.
2. Папфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г. Принципы организации и работы дисплеев на осно-

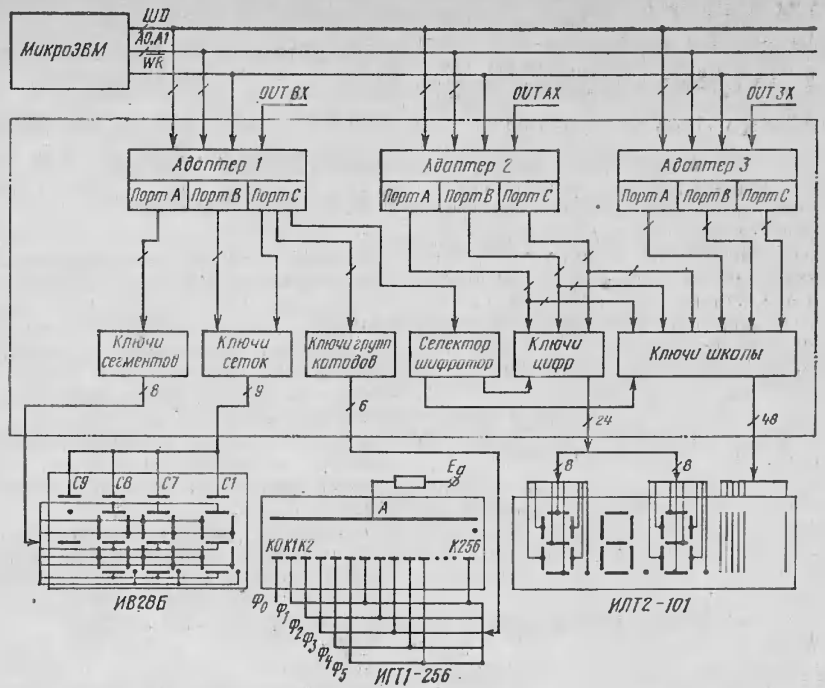


Рис. 2. Структурная схема установки для отображения цифровой и шкальной информации

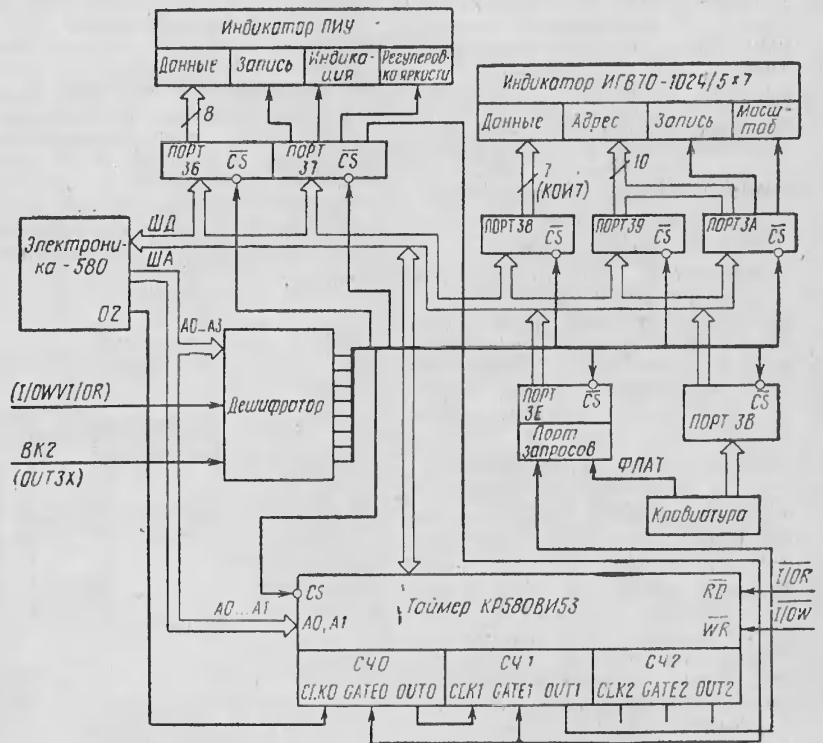


Рис. 3. Структурная схема установки для отображения символьной и псевдографической информации

ве БИС КР580ВГ75 // Микропроцессорные средства и системы.— 1985. № 3.— С. 51—59.

3. Зеленко Г. В. Дисплей для бы-

товой персональной ЭВМ // Микропроцессорные средства и системы.— 1985.— № 3.— С. 60—69.

Статья поступила 23 марта 1987 г.

КЛАВИАТУРА НА БАЗЕ ОДНОКРИСТАЛЬНОЙ МИКРОЭВМ КМ1816ВЕ48

Создание контроллера клавиатуры (КК) на жесткой логике ограничивает его возможности, поэтому целесообразно использовать, например, однокристалльную микроЭВМ КМ1816ВЕ48. Это позволяет при малых аппаратных затратах расширить функциональные возможности контроллера: реализовать два и более регистра на одной клавиатуре, изменить назначение функциональных клавиш клавиатуры командами ЭВМ, в которой используется данный контроллер, использовать буфер хранения набираемых на клавиатуре кодов символов при работе КК в автономном режиме, выдавать коды нескольких символов или слов при нажатии одной клавиши.

МикроЭВМ в КК позволяет также легко организовать связь с удаленной ЭВМ по последовательному интерфейсу, повысить надежность и упростить устройство (рис. 1) в целом по сравнению, например, с блоком клавиатуры 15 ВВ3-97—006 и с блоком клавиатуры, описанным в [1].

Сканирование матрицы клавиш и формирование кодов клавиш, обеспечение протокола ввода-вывода информации и управление индикацией режимов работы выполняет микросхема КМ1816ВЕ48 (DD1).

Однокристалльная микроЭВМ (DD1) содержит 8-разрядный процессор,

РПЗУ команд (1024 байт) с ультрафиолетовым стиранием, ОЗУ (64 байт), 8-разрядный таймер-счетчик и 27 линий ввода-вывода.

Микросхемы DD2, DD3 предназначены для сканирования матрицы клавиатуры по строкам, DD4 — для опроса по столбцам, DD5, DD6 — это буферные элементы для формирования необходимых уровней сигналов интерфейса и индикаторов режимов работы клавиатуры.

МикроЭВМ по линиям порта DB0...DB3 управляет сканированием строк матрицы, а по DB4...DB7 — столбцов клавиатуры, программно контролируя состояние входа таймера-счетчика T1. Появление активного уровня напряжения на входе T1 говорит о том, что нажатие клавиши произошло и после окончания текущего цикла сканирования нажатая клавиша обрабатывается (с последующей выдачей кодов в линию). Коды нажатых клавиш передаются по линии TXD при наличии сигнала готовности TC от ЭВМ.

После каждого цикла сканирования клавиатуры микроЭВМ еще выдает сигнал RC — запрос на прием кодов команд от ЭВМ. Коды команд, приходящие по линии RXD, изменяют коды функциональных клавиш

F0...F9 при использовании клавиатур со стандартным набором клавиш. При этом нажатие одной из функциональных клавиш F0...F9 может генерировать последовательности из нескольких кодов клавиш.

Реализация КК на базе микроЭВМ позволяет программно изменить функциональное назначение отдельных клавиш или клавиатуры в целом и обеспечить гибкий протокол обмена информацией с ЭВМ. Программа заменяется перезаписью содержимого РПЗУ микроЭВМ, а также изменением алгоритма работы под управлением инструкций (кодов команд), полученных от ЭВМ.

С ЭВМ КК связан по последовательному интерфейсу Стык С2 в стартовом режиме (рис. 2).

После поступления сигнала сброса RES микроЭВМ инициализируется, и контроллер приступает к сканированию матрицы клавиатуры. Если нажатия клавиши не произошло, КК выдает запрос на передачу кодов команд со стороны ЭВМ. Тогда при готовности ЭВМ происходит прием инструкции по линии RXD и изменение указанных в команде кодов функциональных клавиш. При нажатии одной или нескольких клавиш КК дешифрует код нажатой клавиши и записывает его в специальный буфер хранения кодов клавиш, синхронизирующий обмен информацией КК с ЭВМ по линии связи.

Максимальное число помещаемых в буфер кодов (20) определяется объемом ОЗУ микроЭВМ. При переполнении буфера ввод кодов блокируется.

Защита от дребезга контактов и обработка одновременного нажатия нескольких клавиш происходят программно.

Затем микроЭВМ анализирует готовность ЭВМ к приему кода клавиши и при активном состоянии сигнала TC выдает в линию очередной код клавиши с учетом содержимого буфера (первый вошедший туда код клавиши первым и выводится). При выдаче очередного кода клавиши из буфера хранения декрементаруется счетчик числа записей, с помощью которого определяется адрес первого вошедшего в буфер кода клавиши.

Индикация режима работы клавиатуры отражает использование функциональных клавиш HP, BP, PUS, LAT, CV.

Длительности битов информационной посылки формируются программно.

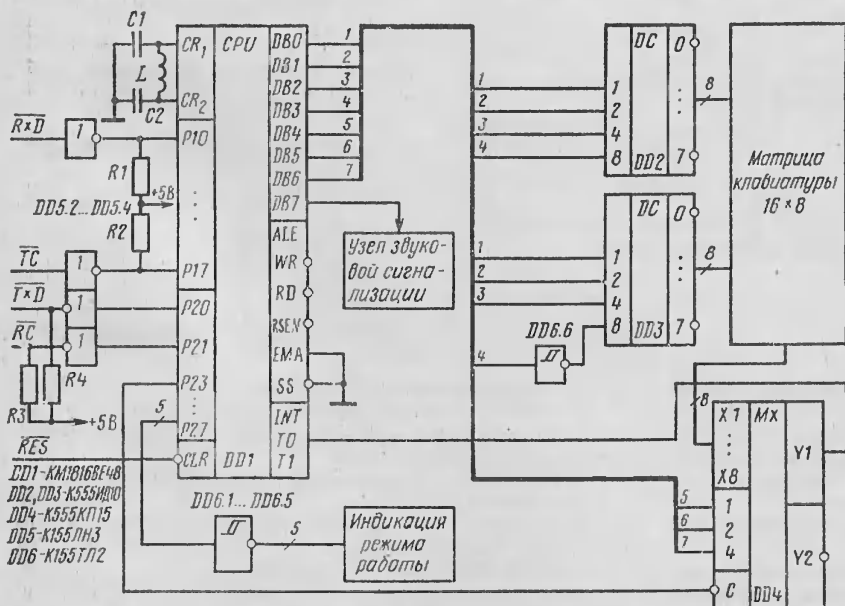


Рис. 1. Функциональная схема контроллера клавиатуры

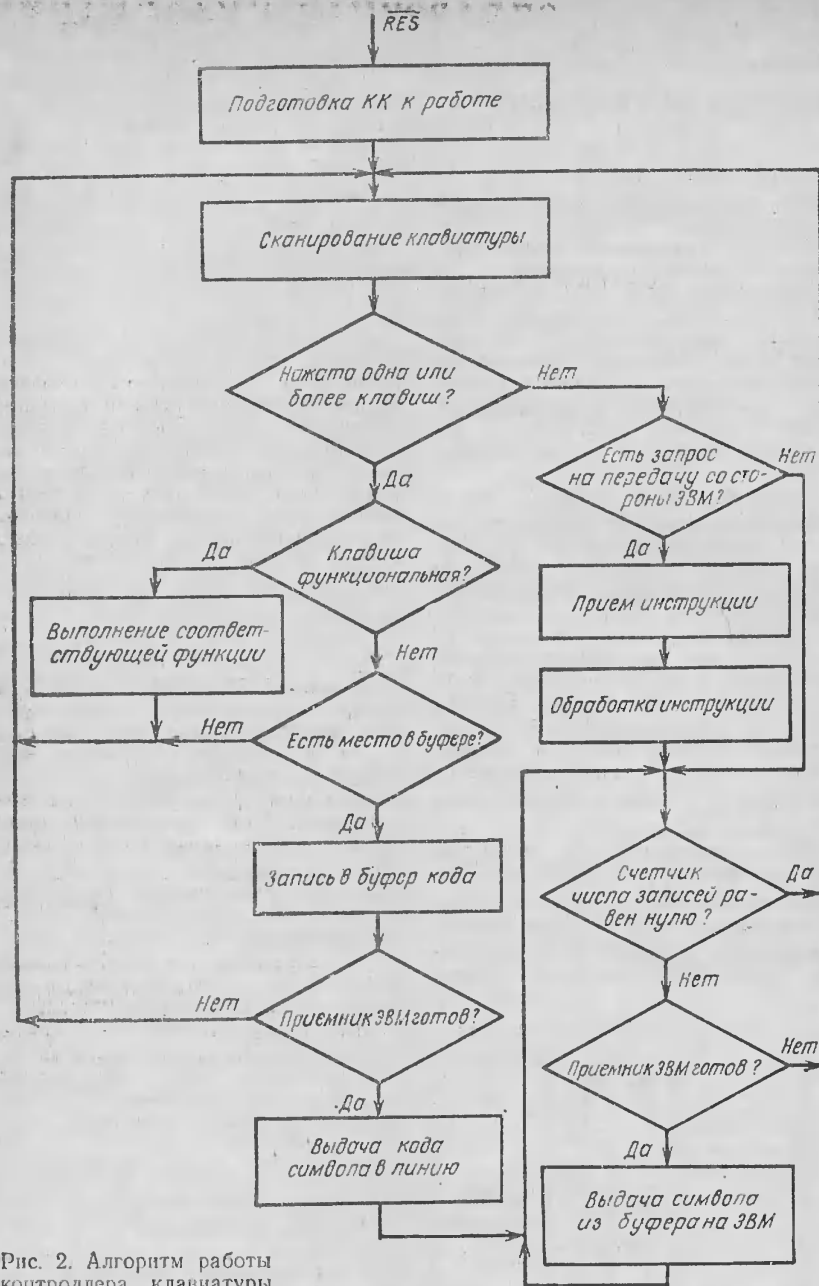


Рис. 2. Алгоритм работы контроллера клавиатуры

но по внутреннему прерыванию от таймера-счетчика, который задает скорость передачи байта. Таким образом, изменяя содержимое таймера-счетчика, можно в широких пределах менять длительность битов, а следовательно, скорость передачи. Обработка прерывания и установив уровень очередного бита, программа сканирует клавиатуру до очередного прерывания. Таким образом, клавиатура постоянно активизирована и готова к работе. Так, при длительности задержки (необходимой для отработки одного бита) равной, например 100 мкс, скорость передачи по последовательному каналу — 9600 бод. Объем памяти команд, использованный для данного КК, — не более 0,6К байт.

Описанный в статье КК выгодно использовать при проектировании любых вычислительных систем, измерительных приборов, средств связи, широкофункциональных терминалов и т. д.

Адрес для справок: 290005, Львов, ул. Ватутина, д. 5; Львовский политехнический институт, тел. 72-88-30.

ЛИТЕРАТУРА

1. Воронин В. И., Макаров К. В., Старшова В. А. Контроллер клавиатуры на базе однокристалльной микроЭВМ KM1816BE48 // Микропроцессорные средства и системы. — 1985. — № 3. — С. 36.
2. Гибсон Г., Лю Ю. Ч. Аппаратные и программные средства микроЭВМ / Пер. с англ. В. Л. Григорьева, под ред. В. В. Сташина. — М.: Финансы и статистика, 1983. — С. 141.

Статья поступила 30 декабря 1986 г.

ИНФОРМАЦИЯ

Философское общество СССР, центральная секция «Методологические проблемы информатики, информатической технологии и информатизации общества» проводит 10—14 октября 1988 года в г. Обнинске Московской области Всесоюзный симпозиум «Методологические и социальные проблемы информатики».

На симпозиуме предполагается работа следующих секций:

1. Концепции информатизации общества.
2. Социально-экономические и правовые проблемы информатизации общества.
3. Развитие новых информационных технологий и их внедрение.

4. Информатизация образования.
5. Информатизация научных исследований.
6. Логические и лингвистические проблемы информационной деятельности.
7. Методологические проблемы когнитивной науки и психология.
8. Круглый стол «Искусственный интеллект и экспертные системы».

Ученые секретари симпозиума:

Воскресенский Анатолий Кириллович, тел. 201-55-04
Семенова Татьяна Николаевна, тел. 408-57-31

Почтовый адрес: 121002, Москва, Смоленский бульвар, 20. Философское общество СССР.

УДК 681.325—181.48

В. В. Давыдовская, В. А. Козловский,
А. В. Малинин, Г. И. Усов

МИКРОСХЕМА АСИНХРОННОГО СТАТИЧЕСКОГО ОЗУ КР537РУ13

Асинхронное статическое запоминающее устройство с произвольной выборкой выполнено по планарной КМОП-технологии с самосовмещенным затвором и оксидной изоляцией элементов. Информационная емкость 4 К бит, организация 1 К×4. Для питания микросхемы используется один источник с напряжением $5 \text{ В} \pm 10\%$. Изделие совместимо с ТТЛ ИС по всем входам и выходам, обеспечивает сохранность информации при питании от источника с напряжением 2 В.

Кристалл ОЗУ заключен в 18-выводной пластмассовый корпус (рис. 1) типа 2107.18-1. Назначение выводов показано в табл. 1.

Кристалл ОЗУ содержит накопитель, выполненный на основе шести транзисторных ячеек памяти, формирователи входных и выходных сигналов, систему усилителей считывания, дешифраторы строки и столбца, разрядную схему и устройство управления перечисленными узлами.

Накопитель ОЗУ организован в виде двух блоков по 2 К бита, которые образуют единую матрицу 64×64 бит. Для выбора четырех из 4096 ячеек памяти на входные формирователи адреса строки и столбца подается 10-разрядный адресный код. Синхронизация работы адресных формирователей осуществляется сигналом \overline{CS} .

Таблица 1
Назначение выводов
ОЗУ КР537РУ13

Вывод	Назначение
1...7	Адресные входы $A_0...A_3, A_7...A_9$
8	Вход сигнала «выбор микросхемы» \overline{CS}
9	Общий вывод U_{GND}
10	Вход сигнала «запись» \overline{WR}
11...14	Входы-выходы информации $D_1/D_01, D_1/D_02, D_1/D_03, D_1/D_04$
15...17	Адресные входы A_4, A_5, A_6
18	Напряжение источника питания U_{CC}

В соответствии с табл. 2 микросхема КР537РУ13 позволяет реализовать режимы записи, считывания, хранения (рис. 2—4). Временные параметры входных сигналов, обеспечивающих правильную работу ЗУ, приведены в табл. 3. Основные электрические параметры показаны в табл. 4.

Асинхронный режим работы обеспечивает микросхемам КР537РУ13 высокие динамические параметры и хорошую помехозащищенность. Рабочая частота БИС серии КР537 сопоставима с биполярными ОЗУ, а потребляемая мощность значительно меньше.

Таблица 2
Режимы работы ОЗУ КР537РУ13

Режим	Входы			Выход D_0
	\overline{CS}	\overline{WR}	D_1	
Запись	L	L	L или H	Z
Хранение	H	X	—	Z
Считывание	L	H	—	L или H

Примечание. L — напряжение «Лог. 0»; H — напряжение «Лог. 1»; X — произвольное логическое состояние; Z — третье состояние (с высоким выходным сопротивлением)

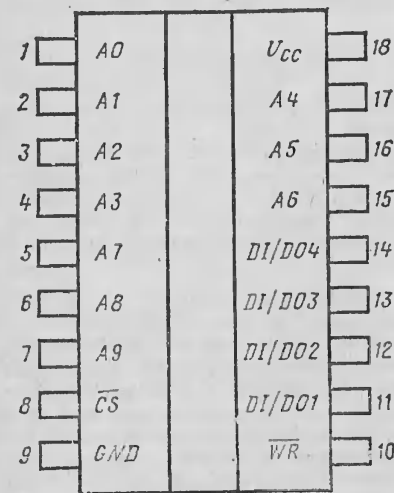


Рис. 1. Условное графическое изображение микросхемы КР537РУ13

дене в табл. 3. Основные электрические параметры показаны в табл. 4.

Асинхронный режим работы обеспечивает микросхемам КР537РУ13 высокие динамические параметры и хорошую помехозащищенность. Рабочая частота БИС серии КР537 сопоставима с биполярными ОЗУ, а потребляемая мощность значительно меньше.

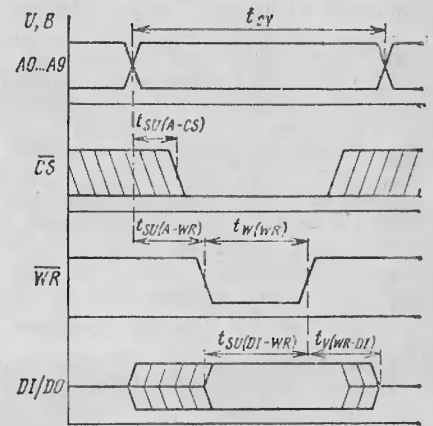


Рис. 2. Режим записи

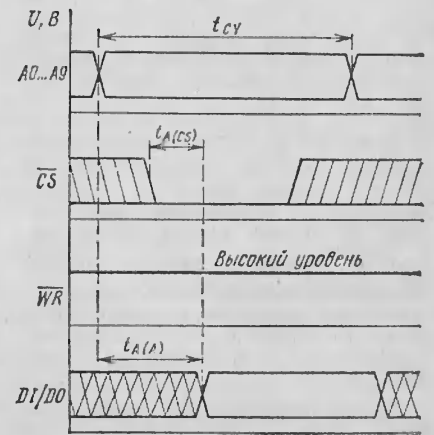


Рис. 3. Режим считывания

Временные параметры входных сигналов

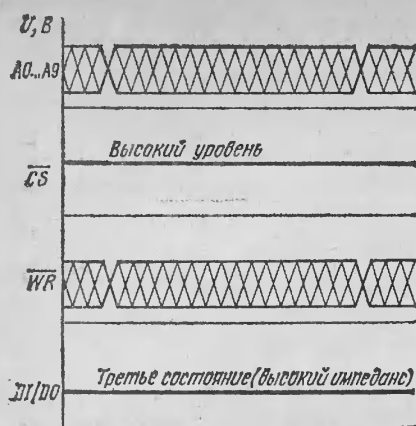


Рис. 4. Режим хранения

Чтение информации может производиться по сигналу адреса, когда $t_{SU(A-CS)} = 0$ и по сигналу \overline{CS} , когда $t_{SU(A-CS)} > 0$. При этом, в случае чтения по сигналу адреса, минимальное значение цикла чтения t_{CY} определяется временем выборки адреса $t_{A(A)}$ и равно ему; в режиме чтения по сигналу выбора \overline{CS} минимальное значение цикла чтения определяется суммой времени выборки по сигналу \overline{CS} $t_{A(CS)}$ и установки сигнала адреса относительно сигнала выбора $t_{SU(A-CS)}$. Время выборки адреса и время выбора \overline{CS} для этих приборов одинаково.

Данные могут записываться раздельно по сигналам \overline{CS} или \overline{WR} и при одновременной подаче этих сигналов. Единственным требованием к временным диаграммам в режиме записи является необходимость задержки сигнала записи (выбора) относительно сигнала адреса $t_{SU(A-WR)}$, $t_{SU(A-CS)}$ и перекрытие сигнала записи сигналом данных на время $t_{SU(DI-WR)}$ большее, чем минимальная длительность сигнала записи $t_W(WR)$.

Когда сигнал \overline{CS} находится в состоянии «Лог. 1», информационный вход принимает состояние высокого импеданса с плавающим потенциалом. В режиме записи, когда сигнал \overline{WR} приходит раньше сигнала \overline{CS} , информационный выход тоже находится в состоянии высокого импеданса в течение всего цикла записи, несмотря на одновременную подачу данных.

При считывании информационный выход переходит из состояния с высоким импедансом в активное состояние по сигналу \overline{CS} и затем через

Наименование	Обозначение	Норма, нс
Время установления сигнала адреса относительно сигнала \overline{CS} , не менее	$t_{SU(A-CS)}$	$\frac{40}{20}$
Время установления сигнала адреса относительно сигнала \overline{WR} , не менее	$t_{SU(A-WR)}$	25
Время установления сигнала данных относительно сигнала \overline{WR} , не менее	$t_{SU(DI-WR)}$	50
Длительность сигнала \overline{WR} , не менее	$t_W(WR)$	110
Время удержания данных относительно сигнала \overline{WR} , не менее	$t_V(WR-DI)$	$\frac{130}{50}$

Примечание. В числителе указаны значения параметров в нормальных условиях, в знаменателе — в диапазоне температур.

Таблица 4

Основные электрические параметры

Наименование, единица измерения	Обозначение	Норма		Примечание
		мин.	макс.	
Напряжение питания, В	U_{CC}	4,5	5,5	
Время цикла считывания (записи), нс	t_{CY}	—	$\frac{160}{200}$	$\frac{25^\circ C}{-10 \dots +70^\circ C}$
Время выборки относительно адреса, нс	$t_{A(A)}$	—	$\frac{160}{200}$	$\frac{25^\circ C}{-10 \dots +70^\circ C}, C_H = 50 \text{ p}$
Время выбора по \overline{CS} , нс	$t_{A(CS)}$	—	$\frac{160}{200}$	$\frac{25^\circ C}{-10 \dots +70^\circ C}, C_H = 50 \text{ p}$
Ток потребления в режиме хранения, мкА	I_{CCS}	—	$\frac{10}{50}$	$\frac{25^\circ C}{-10 \dots +70^\circ C}, E_{П} = c5,5B$
Ток потребления динамический, мА	I_{CCD}	—	$\frac{60}{75}$	$\frac{25^\circ C}{-10 \dots +70^\circ C}, t_{CY} = 235 \text{ нс}$
Напряжение «Лог. 1» сигнала выходной информации при токе нагрузки 2мА, В	U_{OH}	2,8	—	
Напряжение «Лог. 0» сигнала выходной информации при токе нагрузки 4мА, В	U_{OL}	—	0,4	
Минимальное напряжение питания в режиме хранения, В	U_{CCS}	2,0	—	

время, равное времени выбора, на выходе устанавливается истинное значение считываемой информации (0 или 1).

Реализованная схема вывода информации дает возможность осуществлять управление состоянием информационного выхода сигналом \overline{WR} при записи (варьируя момент подачи сигнала относительно \overline{CS}) и сигналом \overline{CS} при считывании,

Считываемая выходная информация D0 имеет ту же полярность, что и входная записываемая информация. Считывание информации происходит без разрушения. Сохранение информации при питании от источника с напряжением 2 В при сверхмалой потребляемой мощности позволяет применять ОЗУ в качестве ЭРПЗУ с высокими динамическими параметрами.

Статья поступила 16 марта 1987 г.

УДК 681.326.35.77

Кобылинский А. В., Калатинец В. М., Заика А. И. Программируемый контроллер клавиатуры и индикации КР580ВВ79 // Микропроцессорные средства и системы. — 1988. — № 1. — С. 3.

Приведены назначение, устройство и принцип работы программируемого контроллера клавиатуры и индикации КР580ВВ79.

UDC 681.326.35.77

Kobylynsky A. V., Kalatynets V. M.; Zaika A. I. Programmable Keyboard and Indication controller КР580ВВ79. // Microprocessor devices and systems.— 1988.— N 1.— P. 3.

General description, block diagram, principles of operation and application notes of the Programmable keyboard and indication controller LSI type КР580ВВ79 are given.

УДК 681.323

Бобков В. А., Горовой В. В., Лаврусенко В. А., Черныковский Д. Н. Одноплатная микроЭВМ на основе БИС серии КР588. // Микропроцессорные средства и системы. — 1988. — № 1. — С. 8.

Описывается опыт создания одноплатной микроЭВМ, построенной полностью на КМОП БИС и БИС серий КР588, КР537, К1623РТ. Совместимость по системе команд и интерфейсу с микроЭВМ «Электроника 60» и ДВК обеспечивает возможность использования этих микроЭВМ в качестве отладочных средств.

UDC 681.323

Bobkov V. A., Gorovoj V. V., Lavrusenko V. A., Chernyakovskiy D. N. Single-board microcomputer using К588 LSI family. // Microprocessor devices and systems.— 1988.— N 1.— P. 8.

Design aspects of single-board microcomputer utilizing only CMOS LSIs of К588, К537 and К1623 families are described. The computer provides compatibility of system interface and instruction set with «Electronika-60» microcomputer, and may be supported by these ones as instrumental tools for program development.

УДК 681.322.1

Иванов В. И., Клепов А. В., Костиков Д. В. Малогабаритные ПЭВМ с жидкокристаллическим экраном — новый класс персональных ЭВМ // Микропроцессорные средства и системы. — 1988. — № 1. — С. 18.

Анализируется зарубежный опыт создания и применения малогабаритных персональных ЭВМ с жидкокристаллическими (ЖК) экранами. Систематизация ПЭВМ с ЖК-экранами наглядно отражает тенденцию развития их схемотехнических и конструкторско-технологических решений.

UDC 681.322.1

Ivanov V. I., Klepov A. V., Kostikov D. V. Compact personal computers with liquid crystal displays — modern type of personal computers // Microprocessor devices and systems.— 1988.— N 1.— P. 18.

The systematic technical characteristics review of modern foreign personal computers with liquid crystal displays is given. Special attention is paid to the development tendencies in the circuit design and technological implementation of such computers.

УДК 681.03.06

Кетков Ю. Л. Графические средства в алгоритмическом языке Бейсик // Микропроцессорные средства и системы. — 1988. — № 1. — С. 25.

Рассмотрены графические компоненты в четырех реализациях языка Бейсик на персональных ЭВМ «Искра 226», IBM PC, MSX и «Электроника 85».

UDC 681.03.06

Ketkov Yu. L. Graphics support in BASIC algorithmic language / Microprocessor devices and systems.— 1988.— N 1.— P. 25.

Graphics support in four BASIC implementations on personal computers «Iskra 226», IBM PC, MSX and «Electronika-85» is discussed.

УДК 681.3.06

Баяковский Ю. М., Михайлова Т. Н., Рыжова И. Г. Реализация базовой графической системы для ПЭВМ // Микропроцессорные средства и системы. — 1988. — № 1. — С. 31.

Рассмотрена реализация базовой графической системы мини-ГКС на Фортране, соответствующая минимальному уровню международного графического стандарта.

UDC 681.3.06

Bayakovskiy Yu. M., Michailova T. N., Ryzhova I. G. Basic graphic support package for personal computer. // Microprocessor devices and systems.— 1988.— N 1.— P. 31.

BASIC graphic support package (mini GKS) implementation in FORTRAN language is described. The package conform to the minimum level of GKS international graphic standard.

УДК 681.325

Дианов А. П., Шелкунов Н. Н. Модули программирования логических схем // Микропроцессорные средства и системы. — 1988. — № 1. — С. 40.

Предложен набор персональных модулей программирования для считывания-записи информации в ПЗУ серий К573, К558, К556 и 2700. Набор предназначен для работы в составе системы проектирования микропроцессорных средств и может использоваться с новыми типами микросхем.

UDC 681.325

Dianov A. P., Shelkunov N. N. PROM programming modules // Microprocessor devices and systems.— 1988.— N 1.— P. 40.

A set of personality modules for К573, К588, К556 and 2700 PROM families programming is proposed. The set is designed to be used in the microprocessor development system and may also be suitable for new PROM families.

УДК 681.323

Кисилевский Я. Ф., Пикусов С. Г. Использование двойной плотности записи на накопителе на гибких магнитных дисках «Электроника ГДМ-7012» при работе с микроЭВМ «Электроника МС 1201» // Микропроцессорные средства и системы. — 1988. — № 1. — С. 49.

Рассматривается возможность использования двойной плотности записи на накопителе на гибких магнитных дисках «Электроника ГДМ-7012» при работе с микроЭВМ «Электроника МС 1201». Показан способ разработки драйвера в ОС RT-11, поддерживающего режим работы с двойной плотностью. Приводится текст программы — аналога аппаратного загрузчика для загрузки ОС с диска с двойной плотностью записи.

УДК 681.325 : 621.397.61

Суранов А. Я., Госьков П. И., Якунин А. Г. Микропроцессорная система измерения параметров бинарных изображений на основе многоэлементного фотоприемника // Микропроцессорные средства и системы. — 1988. — № 1. — С. 54.

Описана микропроцессорная система на основе линейного ПЗС-фотоприемника К1200ЦЛ1, предназначенная для измерения параметров бинарных изображений. Рассмотрено построение основных элементов системы и дано краткое описание их работы.

УДК 681.327.23

Тимофеев Е. Ю. Сопряжение персональных ЭВМ с приборным интерфейсом // Микропроцессорные средства и системы. — 1988. — № 1. — С. 58.

Рассмотрены типовая схема и программное обеспечение, необходимые для сопряжения ПЭВМ с приборным интерфейсом («Каналом общего пользования»). Программное обеспечение ориентировано на МП КР580ИК80 и алгоритмические языки типа Бейсик, Фокал и т. п.

УДК 681.325.5

Торгов Ю. И. Программируемый контроллер клавиатуры КР580ВВ79 // Микропроцессорные средства и системы. — 1988. — № 1. — С. 65.

Характеризуются не только основное назначение БИС КР580ВВ79 (обработка состояний матрицы двухпозиционных датчиков), но и применение в качестве стробируемого порта ввода, 8-уровневого регистра и дополнительного ОЗУ (16 слов × 8 разрядов).

УДК 681.322.1+681.325.5

Панфилов Д. И., Шаронин С. Г., Яковлев С. Е. Сопряжение ЦАП и АЦП с микропроцессорными системами // Микропроцессорные средства и системы. — 1988. — № 1. — С. 72.

Анализируются программно-аппаратные способы сопряжения МПС с ЦАП и АЦП, связанные с особенностями характеристик и режимов работы. Приведено описание устройств, позволяющих изучить и исследовать работу ЦАП и АЦП на практике.

УДК 681.326-181.4

Добуш Ю. Д., Старостенко О. В., Натопта Е. Е. Клавиатура на базе однокристалльной микроЭВМ КМ1816ВЕ48 // Микропроцессорные средства и системы. — 1988. — № 1. — С. 91.

Даны функциональные и технические характеристики контроллера с расширенными функциональными возможностями для работы с клавиатурой на базе однокристалльной микроЭВМ КМ1816ВЕ48, обеспечивающего двусторонний обмен кодами команд, букв, знаков и цифр в коде КОИ-7 по последовательному интерфейсу Стык С2.

UDC 681.323

Kisilevsky Ya. F., Pikusov S. G. Double-density mode in ГМД 7012 floppy disk drive working with «Elektronika-MC1201» microcomputer. // Microprocessor devices and systems.— 1988.— N 1.— P. 49.

Simple method enables double-density operation of ГМД-7012 floppy disk drive when only built-in disk interface of MC1201 microcomputer is available RT-11 DY disk handler modifications needed for such mode are described. Primary bootstrap supporting double-density drive is listed.

UDC 681.325 : 621.397.61

Suranov A. Ya., Gos'kov P. I., Yakunin A. G. Microprocessor system for measurement of binary image parameters using multicell photodetector. // Microprocessor devices and systems.— 1988.— N 1.— P. 54.

Microprocessor system for binary image parameters measurement using linear charge-coupled device (CCD) photodetector K1200ЦЛ1 is described. The structure of main units of the system and their operation are explained.

Timofeev E. Yu. GP-IB interface in personal computer / Microprocessor devices and systems.— 1988.— N 1.— P. 58.

General description of typical hardware and software necessary for connection of personal computer to IEC 625 (GP-IB) interface bus is given. The software is written for КР580ИК80 microprocessor and interpreters such as FOCAL, BASIC etc.

UDC 681.325.5

Torgov Yu. I. Programmable keyboard controller type КР580ВВ79 // Microprocessor devices and systems.— 1988.— N 1.— P. 65.

Standard operation of КР580ВВ79 chip as a scanner of two-state sensor matrix, as well as some non-standard applications, including strobed input port, 8-stage register and external 16×8 bit RAM array are explained.

UDC 681.327.23

Panfilov D. I., Sharonin S. G., Yakovlev S. E. Interfacing ADC and DAC to microprocessor systems. // Microprocessor devices and systems.— 1988.— N 1.— P. 72.

Software and hardware details of ADC and DAC interfacing to microprocessor systems, depending upon their technical characteristics and operation modes are explained. The unit designed for practical examination of ADC and DAC operation in the system is described.

UDC 681.326—181.4

Dobush Yu. D., Starostenko O. V., Natophta E. E. Keyboard controller using КМ1816ВЕ48 single-chip computer // Microprocessor devices and systems.— 1988.— N 1.— P. 91.

Technical features of advanced computer keyboard controller are presented. The controller is built around КМ1816ВЕ48 single-chip computer and performs duplex exchange of command and ASCII character bytes via serial interface С2 (RS232).

МИКРОПРОЦЕССОРЫ В «МАШИНОСТРОЕНИИ»

В интенсификации общественного производства главное место отводится машиностроению. Гезксе сокращение сроков внедрения в производство новых проектно-конструкторских разработок, являющихся необходимым условием повышения их эффективности, возможно лишь при автоматизации технологического проектирования, создании новых автоматических роторных и роторно-конвейерных линий, гибких производственных систем, робототехнических комплексов. Именно поэтому издательство «Машиностроение» большое внимание уделяет выпуску книг по этой тематике.

Важным фактором ускорения научно-технического прогресса является создание и широкое внедрение автоматизированных систем, в том числе систем автоматизированного проектирования (САПР). В книге «САПР: формирование и функционирование проектных модулей» авторы О. Л. Смирнов, С. Н. Падалко и С. А. Пивяский с системных позиций подходят к проектированию САПР как организационно-технической системы, состоящей из пресектных модулей. Каждый проектный модуль они в свою очередь рассматривают как целостную организационно-техническую систему с полным набором компонент (проектная операция, технические средства, информационная среда, организационная система). Приведены методы и алгоритмы, обеспечивающие гибкое формирование проектных модулей на базе САПР. Большое внимание уделено функционированию проектных модулей.

Продолжается выпуск книг в хорошо зарекомендовавшей себя серии «Автоматические манипуляторы и робототехнические системы». Например, в 1987 г. вышла очередная книга под редакцией канд. техн. наук Ю. Г. Козырева и А. А. Кудинсва «Роботизированные производственные комплексы». В книге отражен опыт создания роботизированных технологических комплексов, применяемых для автоматизации процессов изготовления серийно выпускаемых изделий. Рассмотрены вопросы проектирования комплексов, комплектующего оборудования и оснастки, применения промышленных роботов для автоматизации работ в литейном производстве, термическом, кузнечно-прессовом, механообрабатывающих цехах, на сборке, сварке, окраске, на транспортных и других работах.

Проблемы построения систем управления — едва ли не самые трудные при проектировании гибких производственных систем. Коллектив авторов Ленинградского электротехнического института и Технического университета в Дрездене в книге «Управление гибкими произ-

водственными системами: Модели и алгоритмы» под редакцией академика С. В. Емельянова основное внимание уделяют вопросам составления предмодельных описаний материального производства как объекта управления в ГПС и разработке алгоритмических представлений имитационных моделей материальных потоков, структуризации и алгоритмизации задач управления дискретными технологическими производствами, автоматизации промышленных исследований и управления в производстве изделий микроэлектроники. Приведены модели локализации источников неоднородности в технологических процессах группового производства изделий микроэлектроники.

В книге Г. И. Васильева «Автоматизация проектирования металлорежущих станков» рассмотрены вопросы получения и реализации на ЭВМ моделей станка и его узлов, автоматизации расчета, конструирования и испытания механизмов и узлов металлорежущих станков. Приведены алгоритмы выбора оптимальных параметров и структурно-компоновочных решений станочного оборудования.

В издательстве создана новая серия «Гибкие производственные системы» под редакцией академика Н. Г. Бруевича. В этой серии вышла книга Б. Е. Челишева, И. В. Боброва и А. Гонсабес-Сабатера «Автоматизация проектирования технологии в машиностроении». В ней изложены основы автоматизации проектирования технологии изготовления дета-

лей произвольной конфигурации для различных производственных условий и заданий на проектирование. Рассмотрено практическое приложение теории для задач автоматизации проектирования и обосновано построение систем автоматизированного проектирования для разноразных изделий, в том числе новых, ранее не поступивших в производство.

Заслуживает внимания книга В. А. Ратмирова «Управление станками гибких производственных систем». В ней автор рассматривает основные аспекты программного управления станками, вопросы организации иерархических структур. Они анализируются с учетом взаимосвязи задач планирования, управления и контроля обработанных деталей. Приведены сведения по микропроцессорным системам ЧПУ, методам контроля и диагностики, приводам станков. Рассмотрены проблемы организации локальных вычислительных сетей в ГПС.

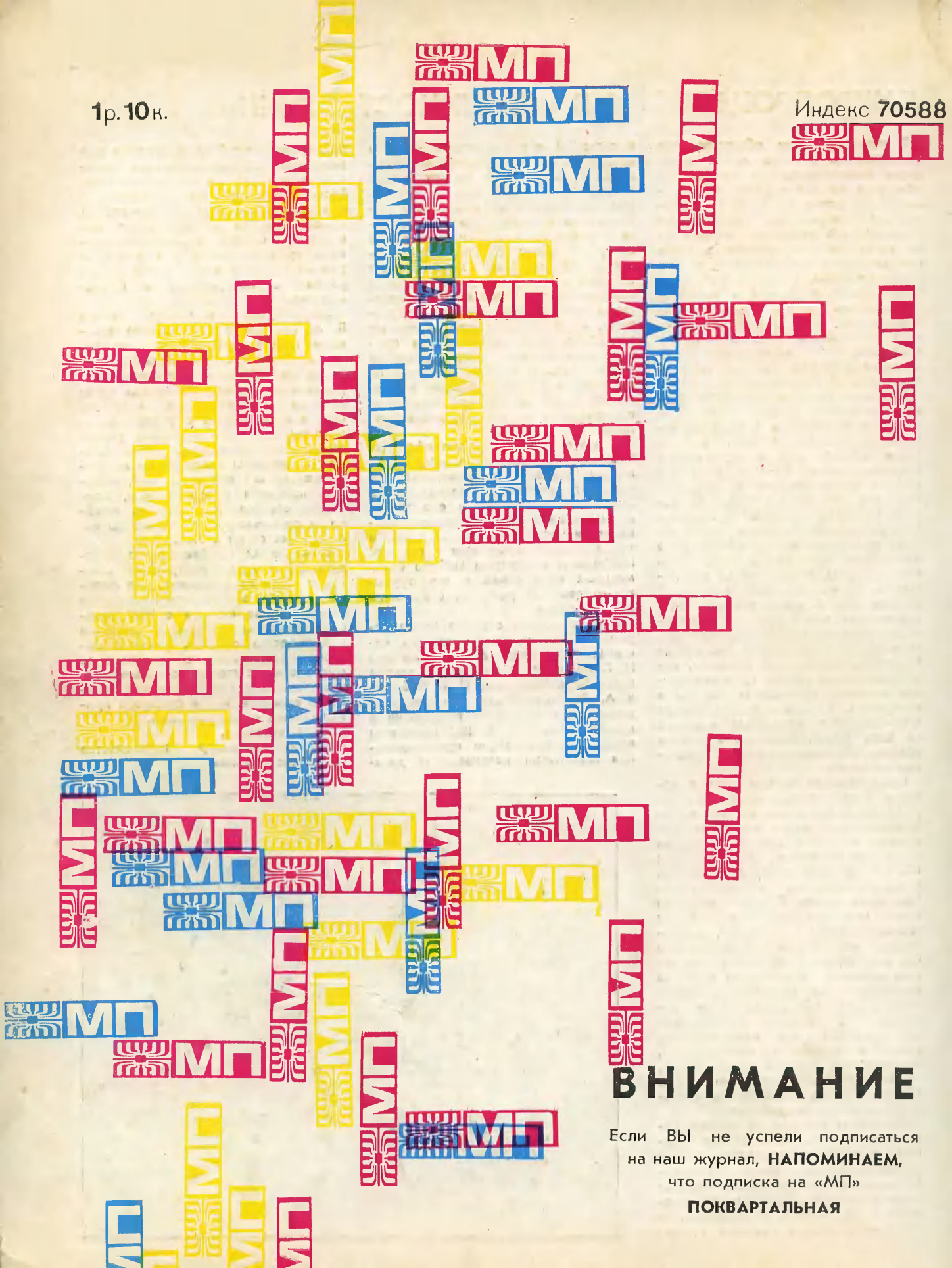
В книге О. И. Аверьянова «Модульный принцип построения станков с ЧПУ» дана комплексная оценка модульного принципа построения многоцелевых станков с ЧПУ, проанализированы тенденции развития производства станков, указаны области рационального применения многоцелевых станков с использованием соответствующего математического аппарата. Книга снабжена приложениями с программами.

Б. И. Пустовалов,
главный редактор
издательства
«Машиностроение»



1р. 10к.

Индекс 70588



ВНИМАНИЕ

Если Вы не успели подписаться
на наш журнал, **НАПОМИНАЕМ,**
что подписка на «МП»

ПОКВАРТАЛЬНАЯ