



МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

5 | 1986

ISSN 0233-4844

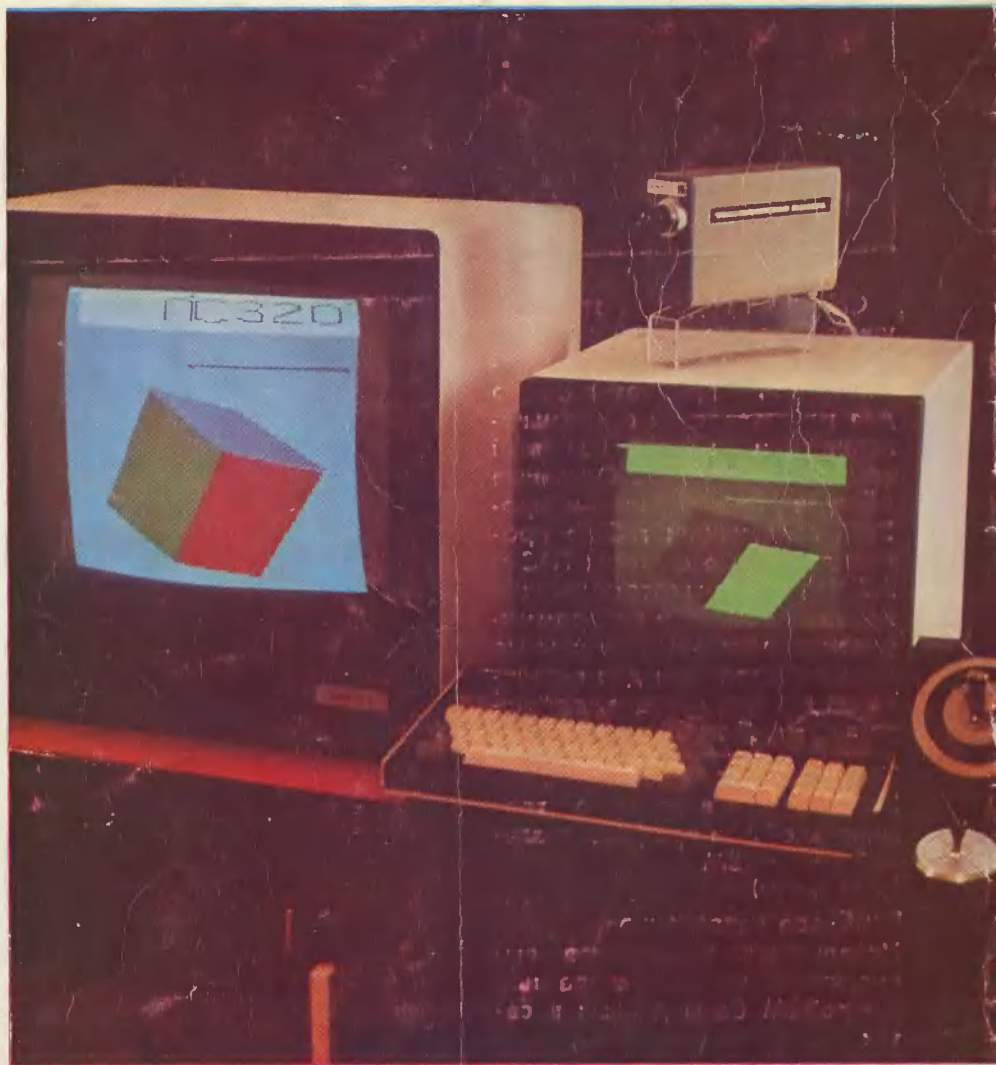
Векторная вычислительная система ПС-320: один миллион операций с плавающей запятой в секунду, 8М байт — ОЗУ

16-разрядные модели микроЭВМ семейства СМ 1800: СМ 1810 и СМ 1814: технические характеристики, программное обеспечение, состав периферийных устройств

Одноплатная микроЭВМ типа ПМВ02 на базе микропроцессорного комплекта К1810: архитектура, организация связи с датчиками и исполнительными устройствами, программное обеспечение

Язык программирования Бейсик/Fs обеспечивает переносимость программ, позволяет эффективно использовать графические, звуковые и игровые возможности различных школьных и бытовых компьютеров

Однокристалльные контроллеры клавиатуры и устройств индикации, выполненные по КМОП-технологии на основе некоммутированных вентиляционных матриц



МИКРОЭВМ СМ 1810

(К ст. Шкамарды А. Н.)



СМ1810 представляет собой универсальную микроЭВМ функционально-модульной архитектуры, предназначенную для использования в промышленной и непромышленной сферах применений. Основные области применения микроЭВМ следующие: гибкие производственные системы ГПС и АСУТП, системы автоматизации проектирования и программирования, локальные и открытые сети, контроллеры для встраивания в оборудование, оргсистемы и сфера обслуживания.

Настольное исполнение

Используется в качестве терминальной или профессиональной микроЭВМ
(фото вверху)

Тумбовое исполнение

Используется в качестве вычислителя, концентратора или микроЭВМ базы данных в сетях
(фото внизу)



ОРГАН
ГОСУДАРСТВЕННОГО
КОМИТЕТА СССР
ПО НАУКЕ И ТЕХНИКЕ

Издается с 1984 года

МПК МИКРО ПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД · НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ · 5/1986 МОСКВА

| | | |
|---|---|----|
| СОДЕРЖАНИЕ МИКРОПРОЦЕССОРНАЯ ТЕХНИКА | Ершов А. П.— Колонка редактора | 2 |
| | Быстродействующая векторная вычислительная система ПС-320 | 3 |
| | Шкамарда А. Н.— Шестнадцатиразрядные микроЭВМ семейства М-1800 | 6 |
| | Долкарт В. М., Степанов В. Н., Крамфус И. Р.— Одноплатная микроЭВМ ПМВ02 на базе МПК БИС К 1810 | 10 |
| | Говорун В. Н., Горбунов Н. В., Мамаков П. В., Рыбаков В. Г., Сытин А. Н., Холоденко Г. М.— МикроЭВМ и управляющие модули на базе БИС серии К 1810 | 13 |
| | Панфилов Д. И., Романенко О. А., Сафанюк В. С., Шаронин С. Г., Шаронин Ю. Г.— Учебная микроЭВМ на основе микропроцессора КМ1810ВМ86 | 16 |
| | Тарутин О. Б.— Интерфейс четырех внешних устройств стандарта ИРПР к микроЭВМ «Электроника 60» | 23 |
| | Бирюков А. В.— Модификация устройства параллельного обмена микроЭВМ «Электроника 60» | 24 |
| | Александров А. Л., Башмакова Е. С., Гуткин М. Л., Либеров А. Б.— О стандарте языка Бейсик | 25 |
| | Корчак А. Е.— Бейсик/Fs — язык программирования школьных и бытовых компьютеров | 31 |
| ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ | Найденов А. В., Романенков В. А.— Программатор ППЗУ на базе микроЭВМ «Электроника К1-20» | 34 |
| | Половинкин И. К.— Кросс-система для микроЭВМ «Электроника ДЗ-28» на базе ЭВМ «Искра 226» | 39 |
| | Жихарев В. И.— Программатор на базе микроЭВМ «Электроника ДЗ-28» | 40 |
| | Глазов А. Б., Костарев С. А., Суханова Е. В.— Эффективные программы умножения для микропроцессора КР5801К80А | 43 |
| | Баран Е. Д., Кошелева Е. И., Салмина Е. М.— Анализатор микропроцессорных систем | 44 |
| | Безобразов В. С., Ларин Б. И., Сохранов В. Ю., Шишкевич А. А.— Отладочный комплекс для 32-разрядной микроЭВМ на основе ДВК-2 | 49 |
| | Тарасов В. В.— Несложная система отладки | 50 |
| | Митрофанов С. П. Пути решения проблемы компьютерной грамотности учащихся северной сельской опорной общеобразовательной средней школы | 52 |
| | Репрограммируемое ПЗУ КМ558РРЗ | 56 |
| | Перельмутер В. М.— Микропроцессорные системы управления электроприводами | 59 |
| Средства отладки | Юрочкин А. Г. Система автоматизации диагностики и контроля МСВТ на базе микроЭВМ «Электроника МС 1212» | 62 |
| | Юрочкин А. Г.— Принципы построения программного обеспечения автоматизированных комплексов диагностики и контроля МСВТ | 64 |
| | Щелкунов П. Н., Дианов А. П.— Универсальный одноплатный микроконтроллер | 65 |
| | Преснухин Л. Н., Соловьев А. Н., Кузнецов Н. Н., Семичастнов О. Л., Разумовский К. П.— Контроллеры индикации и клавиатуры на основе некоммутированных вентиляционных матриц | 70 |
| | Шевченко А. А.— Одноплатный многоцелевой микроконтроллер на базе БИС серии КР580 | 74 |
| | Кишгин С. Б.— Многоголосый электромузыкальный синтезатор | 77 |
| | Дубровский И. И., Финякин Л. Н., Кафаров В. В.— Устройство параллельного ввода аналоговой информации в микроЭВМ | 81 |
| | Васильев С. И., Леонов В. Б.— Сопряжение микроЭВМ «Электроника ДЗ-28» с МПК БИС серии КР580 | 83 |
| | Панфилов Д. И., Шаронин С. Г., Яковлев С. Е.— Организация обмена информацией с микроЭВМ в параллельном коде | 85 |
| | Дмитренко А. П., Накалов Е. Ф.— Контроллер клавишного пульта | 90 |
| Как учить программированию | Злотник Е. М., Стежко И. К., Анищенко В. В., Киркоров С. И.— «Электронный диск» для вычислительных комплексов на базе микроЭВМ «Электроника 60» | 90 |
| | Сорокин Ю. Ю., Лаврентьев В. В., Максимак С. П., Субач В. В.— «Электронный диск» для микроЭВМ «Электроника 60М» | 92 |
| | Внимание! Ошибки перевода и оригинала | 95 |
| | | |
| Справочная информация ПРИМЕНЕНИЕ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| УЧЕБНЫЙ ЦЕНТР | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| Квазидиск | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Главный редактор

А. П. ЕРШОВРедакционная
коллегия:

А. Г. Алексенко
В. В. Бойко
В. М. Брябрин
К. А. Валлев
Г. Р. Громов
(ответственный секретарь)
В. И. Иванов
М. Б. Игнатьев
А. В. Калыев
С. С. Лавров
В. В. Липасев
Б. Н. Наумов
(зам. главного редактора)
С. М. Пеленов
(зам. главного редактора)
А. К. Платонов
Д. А. Поспелов
Ю. А. Чернышев
В. А. Чиганов
И. И. Шагури

Редакционный совет:

Ю. Е. Антипов
Р. Л. Ашастин
Е. П. Велихин
Н. Н. Говорун
В. И. Жильцов
Г. И. Кавалеров
И. И. Малашинин
В. А. Мясников
Ю. Е. Нестерихин
И. В. Прагишвили
Л. Н. Преснухин
В. И. Скурихин
В. Б. Смолов
Ю. М. Соломенцев
В. И. Хохлов
Н. Н. Шереметьевский

Номер подготовили:

Г. Г. Глушкова, В. М. Ларнонова,
С. С. Матвеев, Е. И. Бабич
Фото Ю. А. Бабич и О. В. Чиркина

На первой странице обложки:
Быстродействующая векторная
вычислительная система ПС-320

Корректор Л. С. Глаголева
Технический редактор
Л. А. Горшкова

Адрес редакции: 101820,

проезд Серова, 5, редакция

журнала «Микропроцессорные
средства и системы»

Телефоны 228-18-88; 221-99-26

Сдано в набор 01.09.86

Т17241

Подписано к печати 30.10.86

Формат 84x108¹/₁₆. Бумага № 1.

Высокая печать. Усл. печ. л. 10,08

Уч. изд. л. 15,1. Тираж 40 000 экз.

Зак. 206. Цена 1 р. 10 к.

Орган Государственного комитета

СССР по науке и технике

Московская типография № 13

ПО «Периодика» ВО «Союзполи-

графпром» Государственного

комитета СССР по делам

издательств, полиграфии и книжной

торговли.

107066, Москва, Б-5, Денисовский

пер., д. 30

ВЫЗОВ ПРОГРАММИСТАМ

*Начну с письма читателя Михаила Ивановича Вахатова из Москвы.**...«Не рассчитывая на внимание главного редактора, сообщаю Вам, что я плачу, когда читаю Ваш журнал.**Почему же я плачу, Ваш подписчик с первого номера?**Потому что была надежда найти на Ваших страницах методики, проанализированные обобщения, описания архитектуры конкретных программных систем для ПЭВМ от простых редакторов текста до СУБД и экспертных систем, листингов «готовых» процедур для отечественных ПЭВМ, чем напичканы западные журналы для программирующих пользователей. По ним я, как по лестнице, дошел бы до создания собственной системы или пакетов прикладных программ, отвечающих всем требованиям.**Вы же, начав на мой взгляд правильно, не углубились в этом направлении ни на шаг, объединив под «общей крышей» программирование, схемотехнику и учебный центр, явно направленный на схемотехников.**Контактирующие со мной программисты тоже не довольны последними Вашими номерами. Интеграция «схемотехника — программирование» режет душу. Мой личный взгляд — необходима перестройка Вашей работы».**Тов. Вахатов прав и может полностью рассчитывать на внимание редакции. Нам, как и всем, нужно во многом перестроить нашу работу в том числе и в направлении, подсказываемом этим письмом. Не отрицая роли редакции и редколлегии в организации материала, хочу тем не менее сказать, что еще в большей степени кажущийся перекос в тематике журнала отражает реальное соотношение творческой активности специалистов по вычислительному делу.**Суть перекоса в том, что конструкторы вычислительной техники и программисты по-прежнему остаются в долгу перед страной. Производство интегральных схем опережает степень их использования в массовом производстве нужного разнообразия ПЭВМ и других законченных электронных приборов и изделий. Это искусственно увеличивает объем работы на местах по конструированию печатных плат и установке на них интегральных схем, т. е. того, что тов. Вахатов называет схемотехникой.**Объем схемотехнической работы дополнительно увеличивается из-за недостаточной номенклатуры микропроцессорных наборов и «вспомогательных» СБИСов для управления внешними устройствами, сетевания, машинной графики, связи с реальными объектами и т. п. Недостаток ПЭВМ и других законченных процессоров сужает фронт работы по программным способам реализации функций электронного устройства. Таким образом схемотехники сейчас являются наиболее дееспособной и активной частью специалистов, что и находит свое отражение в структуре журнала.**А что же программисты? Возможно, я буду резок в оценке текущего положения, но критика нам нужна не только в общественной работе.**Наши программисты в целом по-прежнему находятся в плену импортного программного продукта, т. е. либо играют с ним в режиме черного ящика, либо теряют физическое и душевное здоровье, копаясь в исходных кодах в худших традициях реверсной технологии.**Разработчики же оригинальных программ приземляют свою работу, либо зарабатывая деньги родному взгляду, либо находясь в искусственной изоляции в силу ведомственной разобщенности и неразвитости системы научно-технической коммуникации.**В результате общезначимость программистской работы недопустимо низка. Средняя тиражность оконечного программного продукта меньше пяти (!). Книжная полка программиста состоит более чем на 80% из переводной литературы или описаний адаптированных программных изделий, а портфель нашего журнала по программированию тощ и неинтересен.**Необходимо разорвать этот порочный круг, и я призываю коллег-программистов оговзваться делом на письмо тов. М. И. Вахатова,***А. П. Ершов**

УДК 681.322

БЫСТРОДЕЙСТВУЮЩАЯ ВЕКТОРНАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА ПС-320*

Дальнейшее развитие высокопроизводительных вычислительных систем связано с совершенствованием их архитектуры в целях эффективной реализации больших массовых операций и крупных команд-процедур [1]. Пользователь любой вычислительной системы стремится получить возможно более быстрое решение задачи при минимальных затратах на программирование и эксплуатацию. Этого, в частности, можно достичь, если формулировать и выполнять прикладные задачи в содержательных, проблемно-ориентированных терминах структур данных и операций, что, в свою очередь, в значительной мере зависит от уровня машинного языка исполняемой ЭВМ. Поэтому «улучшение» архитектуры вычислительных систем определяется аппаратной и/или микропрограммной реализацией типовых операций и процедур, выполнявшихся ранее программно. Предпосылкой для широкого применения аппаратных средств являются непрерывное увеличение степени интеграции микросхем и снижение их стоимости.

Вычислительная система ПС-320, разработанная в Институте проблем управления совместно с НПО «ЭЛВА», г. Тбилиси, позволяет достичь высокой пользовательской эффективности на основе небольших аппаратных затрат для важного и широко используемого класса задач (линейного программирования, обработки изображений, медицинской и геофизической информации, управления сложными технологическими объектами и научным экспериментом). Основными объектами обработки этих задач являются большие регулярные массивы (векторы и матрицы) символов, целых, фиксированных и плавающих чисел [2].

Для быстрой векторной обработки данных используются два подхода. Первый — присоединение к универсальной ЭВМ дополнительного процессора (как внешнего устройства), выполняющего набор фиксированных операций над массивами (элемент-

ные операции, фильтрацию, обращение матриц, быстрое преобразование Фурье (БПФ) и т. д.) [3, 4]. Достоинством данного решения является возможность использования универсальной ЭВМ в качестве базовой и в какой-то мере накопленного материального обеспечения, а недостатком — ограниченный набор операций, медленный канал обмена информацией с центральным ОЗУ.

Другой подход связан с разработкой векторной ЭВМ, в систему команд которой включены кроме скалярных векторные и матричные операции, а центральный процессор расширен аппаратными средствами векторных операций. Такой подход положен в основу архитектуры вычислительной системы ПС-320, представляющей собой малую векторную ЭВМ с микропрограммным управлением, конвейерным методом обработки массивов, сверхбыстрой внутримашинной магистралью, большим объемом (до 8М байт) оперативной памяти и реализованной на основе элементов высокой степени интеграции (БИС серий К1802, К1804, К565РУ5).

Объединение скалярных и векторных обрабатывающих устройств в одном процессоре позволяет ценой небольших аппаратных затрат обеспечить эффективную обработку и длинных, и коротких массивов данных, совмещать обработку адресов, выборку и запись данных и различные фазы обработки [5].

Структура ПС-320

В системе ПС-320 (рис. 1) управляющий процессор (УП) выпол-

няет процедуры операционной системы. Он инициирует операции ввода-вывода, планирует прикладные и системные задачи, обеспечивает быструю обработку прерываний. Центральный процессор (ЦП), включающий в себя скалярный (СП) и векторный (ВП) сопроцессоры, выполняет прикладные задачи. УП и ЦП через внутримашинную магистраль (ВММ) имеют доступ к общей оперативной памяти (ОЗУ) с блоком базовой арифметики (БА). ВММ обеспечивает быструю блочную передачу 16- и 32-разрядных слов и через системный адаптер имеет выход на стандартную магистраль И41, к которой через периферийные процессоры (ПП) подключаются внешние устройства.

Модульный принцип организации ПС-320 позволяет использовать различные модификации процессора и внешних устройств.

Высокая производительность ПС-320 (до $1 \cdot 10^6$ операций с плавающей точкой в секунду) достигается вследствие конвейерной обработки массивов данных, разделения ЦП на отдельные функциональные устройства, работающие параллельно и в конвейерном режиме, быстрой асинхронной магистрали с совмещением передачи адресов и данных, совмещения ввода-вывода с выполнением массовых операций. Самопроверяемые схемы встроенного контроля, а также системы локального и общего тестового диагностирования повышают показатели надежности и готовности ПС-320.

Архитектура и система команд ПС-320 позволяют не только ускорить выполнение команд, но и существенно упростить прикладные и системные программы. Отдельные программные модули и сегменты данных адресуются независимо, перемещаемы и аппаратно защищены друг от друга, что позволяет организовывать релентные и рекурсивные процедуры. Команды работы с семафорами и очередями обеспечивают синхронизацию и планирование процессов как на одном процессоре, так и на параллельно работающих процессорах (УП, ЦП и ПП). На машинном уровне возможна одновременная работа до 512 процессоров. В программное обеспечение ПС-320 входят кросс-система разработки и отладки программ на языке ассемблера, управляющая система, резидентные ассемблер и компилятор языка Фортран-IV (расширенный векторными командами). Разрабатывается инструментальная операционная система, совместная с ОС ДЕМОС. Кроме того, имеется система разработки и отладки микропрограмм, позволяю-

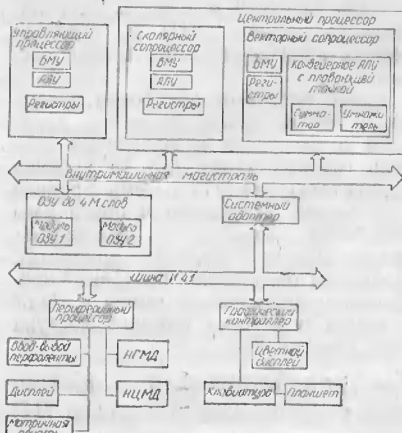


Рис. 1. Структурная схема ПС-320

* Систему разработали: И. В. Прангишвили, А. Н. Вепхвадзе, Л. В. Вейц, Г. Ш. Гудушаури, О. В. Гоголадзе, С. В. Денисенко, И. Ф. Дятчина, В. А. Жуков, Т. И. Зрелова, Я. А. Левертов, В. Д. Малюгин, В. В. Соколов, С. В. Шевцов, А. И. Шкатулла, Р. И. Шубина.

Технические характеристики ПС-320

| | |
|---|---|
| Типы данных | целые числа, числа с фиксированной и плавающей точкой, комплексные числа, строки символов, числовые векторы и матрицы |
| Разрядность данных | 16 бит, 32 бита |
| Объем памяти ОЗУ | 256К слов (слово — 16 разрядов) с расширением до 4М слов — одного слова — 400 не — двойного слова — 800 не |
| Время выборки из ОЗУ с контролем по Хеммингу | |
| Объем памяти микропрограмм: | |
| скалярного процессора | 1К 48-разрядных слов |
| векторного процессора | 1К 64-разрядных слов |
| Пропускная способность внутримашинной магистрали, Мбайт/с | до 10 |
| Тактовая частота процессоров (ЦП, СП, ВП), не | 400 |
| Время выполнения скалярной команды (регистр — регистр), мкс | 2,0 |
| Время выполнения векторной команды, мкс | |
| для фиксированной арифметики: | |
| (16 разрядов) — однооперандные | 0,8 |
| — трехоперандные | 2,4 |
| для плавающей арифметики: | |
| (32 разряда) — однооперандные | 1,0 |
| — трехоперандные | 3,0 |
| Производительность, млн. операций/с: | |
| для фиксированной арифметики | 0,4...1,2 |
| для плавающей арифметики | 0,3...1 |
| по Гибсон III | 0,5 |
| Габаритные размеры центральной части, включающей ЦП, УП, ОЗУ, ВММ и адаптер, мм | 383×399×768 |
| Потребляемая мощность, кВт, не более | 1 |

щая пользователям вводить новые проблемно-ориентированные команды.

Встроенный контроль и системы диагностирования ПС-320

Функциональное диагностирование системы осуществляется с помощью самопроверяемых схем встроенного контроля, конструктивно выполненных в виде отдельных плат в составе процессоров. Эти схемы позволяют обнаруживать неисправности (в момент их первого появления) контролируемых узлов и свои собственные. Кроме этого, в ПС-320 используется локальное и общее тестовое диагностирование. Локальное тестовое диагностирование предназначено для проверки процессоров от «свободные» промежутки времени или по командам операционной системы. Результаты тестирования оцениваются с помощью средств контроля данного процессора. Общие тесты предназначены для проверки исправности всех блоков ПС-320 в режиме профилактики. Результаты проверки сравниваются с эталонами. Информация в ОЗУ хранится в коде Хемминга с самопроверяемым декодером. Код Хемминга позволяет исправлять одиночную ошибку и обнаруживать двойную в считываемом из ОЗУ слове.

Объем дополнительных аппаратных средств на встроенный контроль (не считая ОЗУ) составляет около 15% от общего объема аппаратуры процессоров. При этом используемые в ПС-320 средства контроля и самодиагностирования позволяют сигнализировать о состоянии каждого процессора и отключать отказавший процессор от магистрали. Это дает возможность на базе ПС-320 построить отказоустойчивые системы с различными показателями надежности при введении резервных процессоров.

Взаимодействие процессоров ПС-320

В ПС-320 совместная мультипрограммная работа процессоров обеспечивает высокую производительность при обработке прерываний. Прикладные программы (задачи и процедуры), реализуемые на ЦП, и программы обработки прерывания, входящие в состав управляющей системы (УС) УП, выполняются параллельно.

Управляющий процессор по сигналу прерывания останавливает выполнение текущей задачи на ЦП и обращается к УС с запросом на обработку прерывания. При этом состоянии текущей задачи сохраняется в сегменте задачи, а ее номер записывается в системную очередь запросов.

Управляющая система, выполняющаяся на УП, обрабатывает запросы от блокированных задач, а ЦП аппаратно перекладывается на выполнение очередной готовой задачи. Это повышает загрузку основного вычислительного ресурса — ЦП и, следовательно, производительность всей системы.

Кроме программ обработки прерываний, УС содержит диспетчер, драйверы и загрузчик и формирует очередь готовых задач (до 16 уровней приоритета); на УП восемь уровней прерывания. Диспетчер и программы обработки запросов от прикладных задач имеют низший приоритет. Драйверы организуют послесловный программно-управляемый обмен с ОЗУ и обмен в режиме прямого доступа (без участия УП).

Быстродействующий векторный сопроцессор в составе ЦП

Большинство матричных процессоров, реализованных как периферий-

ные устройства центральной ЭВМ, используются неэффективно из-за того, что вычислитель простаивает, ожидая обмена массива данными с центральной ЭВМ по стандартным, недостаточно быстрым интерфейсным шинам. В ПС-320 передача данных между ВП и центральной ОЗУ осуществляется параллельно с выполнением векторной обработки по ВММ с пропускной способностью 10М байт/с.

Векторный и скалярный сопроцессоры работают совместно. СП выполняет функции управления в рамках ЦП, выбирает команду и запускает микропрограмму ее выполнения. Если это команда управления или скалярной обработки, СП выполняет ее. Если же команда векторная, то СП осуществляет только преобработку и запускает ВП. При выполнении векторной команды СП и ВП работают асинхронно и параллельно. СП формирует и выдает в ОЗУ адреса следующих данных, в то время, как ВП обрабатывает очередные данные на конвейерном АЛУ, реализованном на БИС ТТЛ Шотки.

Структурная схема ВП (рис. 2). Конвейер ВП содержит восемь 32-разрядных регистров общего назначения, восемь 32-разрядных рабочих регистров и семь конвейерных операционных блоков: сложения (три), умножения (один) и универсальные сдвигатели (три). Блок асинхронной связи совместно с буферными регистрами обеспечивает одновременную работу конвейера ВП и обмен данными между ОЗУ и ВП. Запросы на передачу в ВП (или передачу в ОЗУ) очередных компонент векторов данных из микропрограммного устройства управления поступают через блок асинхронной связи в СП, после чего СП инициирует обмен данными между ОЗУ и ВП. О заполненности буферных регистров данными информируют сигналы готовности. ПЗУ перекодировки предназначено для перекодировки кода операции ВП, полученного из СП, в начальный адрес соответствующей микропрограммы и в код настройки ВП, который позволяет аппаратно модифицировать операции, выполняемые отдельными устройствами ВП (например, изменить тип данных и формат данных в операционных блоках). Настройка блока асинхронной связи, а также механизм блокировки выдачи запросов в СП определяют разнообразный тип и структуру данных: одномерный или двумерный массив, вектор или скаляр, одно или два слова.

Базовыми операциями ВП являются векторные редуцирующие и покомпонентные логические и арифметические операции, включающие в себя: вычисление, сравнение, упорядочение, классификацию, поиск по максимуму, минимуму и интервалу, сдвиг, сортировку, сжатие, индексацию и другие операции над массивами. Кроме того, ВП выполняет специальные

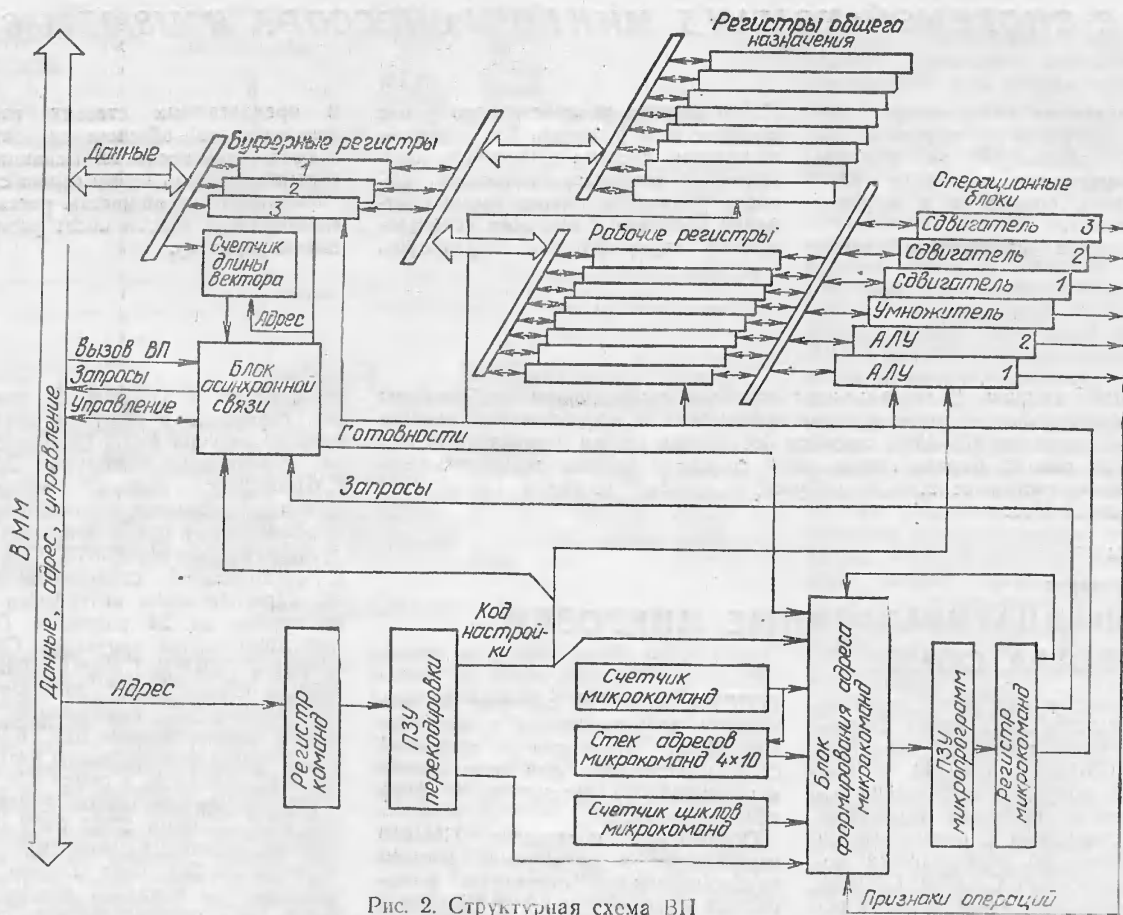


Рис. 2. Структурная схема ВП

операции: прямое и обратное БПФ, вычисление полиномов и спектральных функций.

Такт работы обрабатывающего конвейера составляет 400 нс, а объем аппаратуры, включающей БИСы 8-разрядных сумматоров и 12-разрядных умножителей серии К1802, — четыре печатные Европлаты.

Система микропрограммирования ВП. Микропрограммное управление ВП реализовано на основе БИС серии К1804 с совмещением выполнения текущей микрокоманды и считывания следующей. Объем ПЗУ — 1024×64 бит. Для управления каждым слоем вычислителя ВП, организованного по конвейерному принципу, отводится своя часть микрокоманды. Высокий уровень операций вычислителя, возможность инициализировать их параллельную работу, а также многофазная реализация микрокоманды, задающая порядок выполнения операций в микрокоманде, в том числе при их обращении к одному и тому же ресурсу, позволяют реализовать достаточно сложные микрокоманды и обеспечивают хорошую гибкость при микропрограммировании различных конвейерных алгоритмов.

Возможно введение новых проблемно-ориентированных команд в ВП.

Разработан микроассемблер, максимально избавляющий пользователя от знания тонкостей аппаратуры, отличающийся свободным форматом и укрупненным видом записи операций в микрокоманде. В микрокоманде может быть задано до 10 функционально законченных операций, например: арифметико-логические операции, умножения 24-разрядных чисел, сдвиги арифметические, логические, циклические 16-разрядных чисел на произвольное число разрядов, операции нормализации 16-разрядных чисел, операции со стеком микрокоманд, переходы в два произвольных адреса управляющей памяти, многонаправленные переходы по значению регистра, операции, обеспечивающие организацию в микропрограмме циклов и подмикропрограмм. При этом во всех арифметико-логических операциях вырабатываются признаки, доступные операциям перехода.

Кросс-система отладки включает в себя транслятор, интерпретатор и программу выхода кода микропрограмм на ленту в формате, необходимом для промышленного программатора.

Базовый набор микропрограмм, реализующий 77 команд над векторами целых и плавающих чисел, а также функциональный тест, занима-

ет 512 ячеек ПЗУ. Остальные 512 ячеек отведены под проблемно-ориентированные микропрограммы обработки матриц, комплексных чисел, вычисления элементарных функций в БПФ.

Телефон для справок: 334-96-61 (г. Москва), Малюгин В. Д.

ЛИТЕРАТУРА

1. Аксенов В. П. и др. Структура и характеристики высокопроизводительных ЭВМ и систем // Зарубежная радиотехника. — 1982. — № 4.
2. А. с. 1120340 (СССР). Управляющая векторная вычислительная система // Открытия. Изобретения. — 1984. — № 39. — С. 150.
3. Толстых Б. Л. и др. Быстролетящий периферийный процессор «Электроника МТ-70» // УСИМ. — 1983. — № 4. — С. 122—125.
4. Бродский И. И. Высокопроизводительный периферийный векторный процессор А-12 // Автоматика. — 1984. — № 4. — С. 29—35.
5. Малюгин В. Д. и др. Архитектура вычислительной системы ИС-320: Сб. статей Всесоюзной научно-технической конференции «Проблемы создания и использования мини- и микроЭВМ» в г. Вильнюсе. 1985. — М.: ИНЭУМ, 1985. — С. 18—20.

Статья поступила 4 февраля 1986 г.

В приведенных ниже четырех статьях рассматриваются варианты исполнения микроЭВМ на базовом микропроцессорном комплекте БИС серии К1810, описанном в журнале № 1 за 1986 г.

Программное обеспечение серии

К1810 широко распространено у нас в стране и за рубежом. Его основа — модульная система подготовки программ на ассемблере и языках высокого уровня, а также система отладки программ с широким использованием внутрисхемных эмуляторов,

В предлагаемых статьях вопросы программного обеспечения затрагиваются лишь косвенно. Редакция надеется посвятить этому серию статей.

Несмотря на общность рассматриваемой темы, статьи носят разноплановый характер.

От редакции. С этого номера журнала мы начинаем публикацию цикла статей, посвященных описанию технических и программных средств микроЭВМ семейства SM 1800, способов построения систем различного назначения на их основе. Первая статья цикла содержит краткие технические характеристики, состав основных модулей, способы поставки микроЭВМ SM 1810 и SM 1814.

УДК 681.324

А. Н. Шкамарда

ШЕСТНАДЦАТИРАЗРЯДНЫЕ МИКРОЭВМ СЕМЕЙСТВА SM1800

Заканчивается подготовка серийного производства двух новых 16-разрядных моделей микроЭВМ семейства SM1800: SM1810 и SM1814. В этих микроЭВМ получены качественно новые значения системных характеристик по сравнению с выпускаемыми промышленностью 8-разрядными моделями микроЭВМ SM1800, SM1804. Применение новых архитектурных решений, использование современной элементной базы, введение новых периферийных устройств, в том числе дисков типа Винчестер, определили достаточно высокий технический уровень разработок микроЭВМ.

SM1810 представляет собой универсальную микроЭВМ функционально-модульной архитектуры [1], предназначенную для использования в промышленности и непромышленной сфере применения. Основные области применения микроЭВМ следующие: гибкие производственные системы

(ГПС) и АСУ ТП, системы автоматизации проектирования и программирования, локальные и открытые сети, контроллеры для встраивания в оборудование, оргсистемы и сфера обслуживания.

Особенность построения SM1810 заключается в расширении магистрально-модульной структуры 8-разрядных моделей семейства SM1800 и обеспечении аппаратной и программной совместимости с ними. Эволюционное развитие семейства микроЭВМ позволило одновременно с началом освоения выпуска SM1810 иметь в ее составе средства передачи данных, отображения и устройства связи с объектом (УСО), что существенно для пользователей. Общая структура микроЭВМ SM1810 приведена на рисунке.

SM1810 состоит из ряда вновь разработанных модулей и устройств, а также выпускаемых серийно в соста-

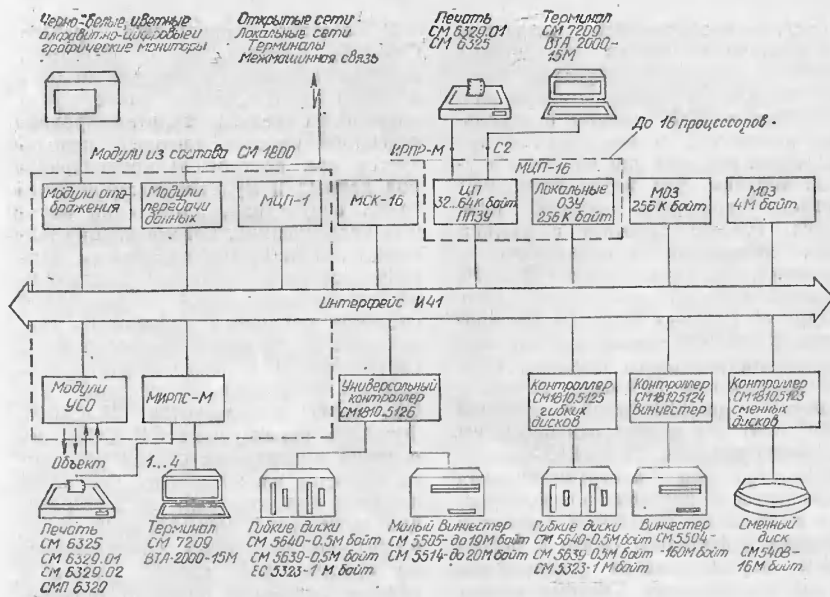
ве SM1800 и SM1804. В структурах совместно с модулем центрального процессора МЦП-16 используется 8-разрядный модуль МЦП-1 SM1800.2202, который выполняет функции процессора ввода-вывода и обеспечивает взаимодействие с модулями из состава SM1800. В связи с организацией страничной памяти, адресная шина интерфейса И41 расширена до 24 разрядов. Основным компонентом структуры SM1810 является МЦП-16 SM1810.2204. Этот модуль обрабатывает логическую и арифметическую информацию, выполнен с применением БИС К1810 и обеспечивает формирование интерфейсов И41, ИРПР-М и стыка С2. В составе модуля локальное двухходовое ОЗУ объемом 256К байт с коррекцией, перепрограммируемое ПЗУ объемом до 64К байт, таймер, БИС ввода-вывода и разъем для подключения БИС арифметического процессора. МЦП-16 обеспечивает подключение периферийных устройств, печати и дисплея, не занимая интерфейса И41, а в случае автономного применения может выполнять функции локальной микроЭВМ. Общее число МЦП-16 в системе — до 16. Арбитраж внутренней шины И41 осуществляется модулем системного контроля МСК-16 SM1810.2005, в составе которого нагрузочные элементы интерфейса и электрическая схема передней панели управления микроЭВМ. Тип арбитража последовательный или циклический.

В составе SM1810 два варианта системного ОЗУ: МОЗ SM1810.3515 — объемом 256К байт с коррекцией ошибок — выполнен на схемах К565РУ5А с внутренним объемом элементов памяти 16К бит в корпусе; МОЗ SM1810.3516 — объемом 4М байт с коррекцией ошибок — состоит из блока управления и четырех накопителей (каждый объемом 1М байт). Различные комбинации указанных модулей ОЗУ позволяют компоновать системы разной степени сложности и производительности. Модули МЦП-16, МСК-16, МЦП-1 и МОЗ составляют вычислительное ядро SM1810.

Основным устройством ввода-вывода общесистемной информации является накопитель на миниатюрном гибком диске ИГМД. В SM1810 при-

Технические характеристики

| Система команд определяется архитектурой микропроцессора К1810. | |
|---|---|
| Ширина шины адреса | 24 бит; |
| Ширина шины данных | 16 (8) бит; |
| Производительность центрального процессора МЦП-16 | до 1 млн. операций/с; |
| Непосредственная адресация | до 1М байт; |
| Объем страничной памяти | до 16М байт; |
| Внутренний интерфейс | И41 |
| Основа элементная база | комплекты КР550 и К1810 |
| Интерфейсы связи | ИРПР, ИРПР-М, ИРПС, ИЛПС, ТГ, стыки С2 и С1 |
| Средства связи с объектом | аналоговый и дискретный ввод-вывод |
| Средства отображения | алфавитно-цифровая и графическая информация в черно-белом и цветном отображении |
| Наличие мультипроцессорного режима работы | до 16-и процессорных модулей |
| Программное обеспечение | шесть операционных систем |
| Конструктивные исполнения | настольное (см. с. 2 обложки журнала), в стойке, встраиваемое в оборудование. |



Структура SM1810

меняются три типа таких накопителей: SM5640 (производство ГДР) с дискеттами односторонней записи плотностью 80 дорожек и емкостью до 0,5М байт; SM5639 (ПНР) с дискеттами двусторонней записи плотностью по 40 дорожек, общей емкостью 0,5М байт; SM5323 (НРБ) с дискеттами двусторонней записи плотностью по 80 дорожек, общей емкостью 1М байт. Кроме того, используются накопитель на сменном диске SM5408 (СССР) емкостью 16М байт с двумя носителями (сменный и фиксированный) и диски типа Винчестер SM5504 (ГДР) емкостью 20...160М байт (в зависимости от количества подклученных поверхностей магнитных носителей). С применением устройств SM5408 и SM5504 расширяются системные возможности SM1810 при организации баз данных в системах. В локальных сетях эти устройства могут применяться в составе «файлсервера».

Универсальный контроллер дисков (см. рисунок) используется в SM1810 при выполнении функций профессиональных ЭВМ и автоматизированных рабочих мест (АРМ). В качестве консольного устройства оператора микроЭВМ SM1810 служат дисплеи ВТА-2000-15М (СССР) или SM7209 (ПНР), позволяющие отображать русские и латинские, строчные и прописные буквы и цифры, а также простейшую псевдографику. Подключение их к SM1810 осуществляется через интерфейс ИРПС, стык С2 модулей МИРПС-М или МЦП-16. К дисплею SM7209 можно подсоединять печатающее устройство и использовать его, например, в системах для обработки текстов.

Поскольку SM1810 рассчитана на различные применения, в ее состав

входит несколько типов печатающих устройств. Растровые устройства печати SM6329.01 и SM6329.02 (ГДР) позволяют выводить соответственно 80 и 136 знаков в строке (растр печати 9×9, скорость печати — 100 знаков/с, имеется возможность передней закладки бумаги). Интерфейсы связи ИРПС, ИРПС и ИРПС-М. В устройстве печати SM6325 (ПНР) растр может быть расширен до 9×n (где n — произвольное число). В состав SM1810 входит также растровое печатающее устройство SM16320 (СССР). Все эти печатающие устройства предназначены для общесистемных применений. Для тех систем, где требуется качественная печать, могут использоваться устройства SM6317 (ГДР) и SM6326 (НРБ) типа «Ромашка» при скорости печати 30 знаков/с.

Для вывода алфавитно-цифровой и графической растровой информации используются модули МВСТ SM1800.7003 и МВТ SM1800.7004 из состава SM1800, позволяющие подключать мониторы А543-14, ВК51Ц61 или бытовые телевизоры. Таков основной состав модулей управления и периферийных устройств, используемых в SM1810.

Построение распределенных и сетевых конфигураций с машинами SM ЭВМ, ЕС ЭВМ, а также зарубежными ЭВМ со стандартизованными интерфейсами связи осуществляется с помощью: модуля связи с модемом МСМ SM1800.8501, реализующего стык С2; четырехканального МИРПС-М SM1800.4106; микропроцессорного модуля МС SM1800.8519, реализующего программируемый протокол и выход в сети; модуля связи с телетайпом МСТ SM1800.8504; микропроцессорного модуля для организации локаль-

ной сети промышленных применений МИЛПС SM1800.4506; программируемого канального адаптера ПКА SM1800.8527 для связи с телемеханикой; встраиваемого модема 600/19200 для организации межмашинной связи типа «многоточка» на расстоянии до 30 км.

Для применений SM1810 в промышленности или в научных исследованиях используются модули и устройства связи с объектом УСО, в том числе модули ввода-вывода дискретной и аналоговой информации. Модуль ввода дискретных сигналов МВВД SM1800.9302 предназначен для ввода сигналов от дискретных датчиков; имеет 16 каналов, гальваническую развязку между цепями управления и цепями датчиков. Модуль вывода дискретных сигналов МВД SM1800.9303 используется для вывода на исполнительные механизмы сигналов двухпозиционного регулирования; имеет 8 каналов, максимальную частоту выходных сигналов 10 кГц, гальваническую развязку между цепями управления и исполнительными цепями. Модуль ввода число-импульсных сигналов МВВЧ SM1800.9304 имеет 2 канала ввода, разрядность счетчика по каждому каналу 8 двоичных разрядов, максимальную частоту входных сигналов 20 кГц, гальваническую развязку. Модуль вывода дискретных сигналов повышенной мощности МВДМ SM1800.9701 применяется для коммутации исполнительных устройств без промежуточных усилителей; имеет 4 канала, максимальную частоту 100 кГц, защиту от перегрузок по току и гальваническую развязку. Модуль управления цепями переменного тока повышенной мощности МУПТ SM1800.9702 используется для бесконтактного управления исполнительными механизмами, например, реле, мощными ключами, контакторами и т. д.; имеет 4 канала, максимальный ток коммутации 3 А, гальваническую развязку.

В настоящее время закончена разработка универсального микропроцессорного модуля дискретного ввода-вывода УМДВВ SM1810.9308. Этот модуль обеспечивает ввод, вывод и обработку дискретных сигналов, ввод и обработку число-импульсных сигналов, вывод широко-импульсных сигналов, а также может выполнять функции модуля сбора и обработки инициативных сигналов. Модуль имеет индивидуальный последовательный канал ввода-вывода информации. УМДВВ позволит расширить функции управления SM1810 в области обработки дискретной информации (например, в машиностроении).

Для ввода аналоговых сигналов от датчиков и преобразования их в двоичный код разработаны 2 модуля. Первый — МВВА SM1800.9201 — имеет диапазон входных сигналов ±5...0 В, 16 каналов однопроводной

коммутации или 8 двухпроводной, максимальное время преобразования 100 мс, 13 двоичных разрядов преобразования. Второй — быстродействующий МВВА-1 СМ1800.9204 — имеет диапазоны измерения $\pm 1...0$ В, $\pm 5...0$ В, 32 однопроводных канала, максимальное время преобразования сигналов по одному каналу 80 мкс, 12 разрядов преобразования. При необходимости подключения более 32 входных каналов совместно с МВВА-1 используются модули коммутации аналоговых сигналов МКАС СМ1800.8517 или СМ1800.8518. Первый позволяет коммутировать 64 однополосных каналов со временем преобразования 1 мкс, второй — 16 двухполосных канала со временем преобразования 100 мкс. Для преобразования токовых сигналов в сигналы напряжения и подавления помех нормального вида используются четыре различных модуля нормализации СМ1800.9211 01—04. Питание аналогового ввода осуществляется от модуля изолированного питания МИП СМ1800.0303 или от модуля аналогового питания МАП СМ1800.0302. Входные напряжения с программируемым уровнем напряжения уставки сравниваются модулем компаратора уровня МКУ СМ1800.9203 (входных каналов — 8, диапазон измерения $\pm 5...0$ В, 10 разрядов цифрового кода уставки, время преобразования не более 15 мкс). Исполнительные воздействия выдаются модулем аналогового вывода МАВ СМ1800.9202, предназначенным для преобразования двоичного кода в постоянное напряжение или постоянный ток. Имеет 10 разрядов преобразования, диапазон входных сигналов 0...10 В или 0...5 мА, время преобразования не более 10 мкс, 2 канала.

Таким образом, номенклатура основных модулей УСО обеспечивает применение СМ1810 в системах управления.

В составе СМ1810 используются специализированные модули и устройства, в том числе для организации быстрой связи между внутрисистемными интерфейсами И41—И41, И41—«общая шина». Следует отметить, что в составе СМ1810 может быть применено более 60 модулей и устройств, позволяющих строить гибкие управляющие структуры микроЭВМ [2].

Программное обеспечение микроЭВМ СМ1810 достаточно развито. В составе СМ1810 имеются следующие системы программного обеспечения (ПО): тестовые; инструментальные; исполнительные; прикладные. Поскольку СМ1810 универсальная микроЭВМ и предназначена для различных применений, структура ПО строится с ориентацией на определенные направления применений.

Тестовая операционная система ТОС-86 обеспечивает проверку и диагностику неисправностей всех средств, используемых в СМ1810.

В системе предусмотрены адаптация под различный состав технических средств и открытая структура. Имеются начальные тесты, которые размещаются в ПЗУ модуля центрального процессора и запускаются при включении питания или нажатии клавиши «сброс». Эти тесты позволяют провести первичную проверку комплекса. ТОС-86 работает с различными дисковыми накопителями в формате и под управлением БОС1810, описание которой приводится далее.

Одно из главных направлений применения СМ1810 составляют системы управления реального времени. Поэтому разработка ПО была начата с дисковой операционной системы (ДОС1810) для подготовки программ реального времени. ДОС1810 — однопользовательская инструментальная операционная система, обеспечивающая выполнение обработки данных и действия с файлами. Связь с оператором осуществляется посредством устройства консоли. Система может работать в диалоговом и пакетном режимах. ДОС1810 имеет две системы программирования: для 8- и 16-разрядных микропроцессоров. Она обеспечивает отладку программ реального времени, в том числе прикладных программ, работающих под управлением операционных систем реального времени микроЭВМ СМ1800, СМ1804 и СМ1810; представляет совокупность программ, реализующих отдельные функции, необходимые при подготовке программ пользователя (ее основной программой является супервизор, обеспечивающий файловый доступ ко всем периферийным устройствам микроЭВМ); обеспечивает редактирование исходных текстов программ. Перемещаемые объектные модули, которые получены с помощью трансляторов из программ, написанных на различных языках программирования, имеют идентичную структуру. Это позволяет объединять разноязыковые программы, используя компоновщик системы. Настройщик системы ориентирует компоновочную программу на загрузочный адрес, после чего программа становится готовой к загрузке и выполнению. Возможно выполнение программ под управлением отладчика.

ДОС1810 реализует следующие функции: обслуживание накопителей на различных дисках; создание файлов и доступ к ним; редактирование текстов программ; программирование для 8-разрядного микропроцессора на языках макроассемблер-80, ПЛ/М-80, Фортран-80, Бейсик-80; программирование для 16-разрядного микропроцессора на языках макроассемблер-86, ПЛ/М-86, Фортран-86, Паскаль-86; автоматическую компоновку объектных модулей; загрузку и выполнение программ; отладку программ; ведение библиотеки стандартных и объектных модулей; подготовку документации машинным способом.

В составе технических средств СМ1810, работающих с ДОС1810, должны находиться модули МЦП-1 и МЦП-16. В СМ1810 может быть реализован режим мультимикропроцессорной работы, который используется при построении концентраторов данных и мультиплексоров в сетевых структурах, проведении научных исследований, автоматизации технологических процессов. Режим многопроцессорной работы поддерживается исполнительными операционной системой реального времени со специализацией функций процессоров ОС СФП СМ1810. В качестве инструментальной операционной системы для ОС СФП используется ДОС1810. Функции, реализуемые ОС СФП: генерация прикладных систем реального времени из компонент ОС СФП и задач пользователя; диспетчеризация вычислений, межзадачный обмен и синхронизация задач; обслуживание по приоритетной схеме; работа в режиме реального времени; межпроцессорный обмен; обслуживание нескольких терминалов и печатающих устройств; межмашинный обмен; обслуживание системы ввода-вывода. Подготовка прикладных программ осуществляется на языках макроассемблер-86, ПЛ/М-86, Фортран-86. Генерация системы выполняется с использованием средств компоновки и настройки ДОС1810.

Общесистемные применения СМ1810 в оргсистемах, текстовых системах, системах обработки экономической и деловой информации поддерживаются ОС Микрос-86. Это однопользовательская инструментальная операционная система общего назначения открытого типа, позволяющая выполнять действия с файлами, поддерживать работу с ОЗУ емкостью до 1М байта, накопителями на дисках различных типов. Микрос-86 реализует функции: диалог оператора с микроЭВМ; связь с внешними устройствами; отладку программ в режиме диалога; программирование на языках ассемблер-86, Фортран-86, Паскаль-86, Бейсик-86, СИ-86; ввод и редактирование текстов; компиляцию, загрузку и выполнение программ; поддержку работы с квазидиском; включение нестандартных устройств и разнотипных дисковых накопителей. Микрос-86 функционально совместима с ОС1800, СР/М-86 и снизу вверх с ССР/М-86.

Решение задач реального времени поддерживается еще одной универсальной большой операционной системой реального времени БОС1810. Эта многопользовательская, многопрограммная операционная система предназначена для построения на ее основе прикладных систем, исполняемых на микропроцессоре серии К1810. БОС1810 отличается от других существующих систем реального времени СМ1810 и СМ1800 тем, что предоставляет пользователям как ис-

полнительные, так и инструментальные средства для редактирования, трансляции, компоновки и настройки прикладных программ. Языки программирования БОС1810: микроассемблер-86, ПЛ/М-86, Паскаль-86, Фортран-86, СИ.

БОС1810 имеет объектно-ориентированную архитектуру, которая основывается на модульном принципе построения и дает возможность пользователю манипулировать этими модулями, называемыми объектами, а не постигать все тонкости, заложенные в ОС. Преимущество такой архитектуры заключается в сокращении времени написания прикладных программ, так как полное изучение БОС1810 достаточно трудоемко. БОС1810 реализует функции диспетчеризации, обеспечивая первоочередное выполнение задач, имеющих больший приоритет. Задачи могут координироваться между собой тремя способами: обмениваться информацией, взаимноисключать друг друга и синхронизироваться. БОС1810 является конфигурируемой системой, поддерживает динамическое распределение памяти и обеспечивает настройку на адрес во время загрузки.

В составе СМ1810 имеется операционная система общего назначения, ориентированная на программную поддержку функционирования в качестве профессиональной персональной микроЭВМ. С помощью малой операционной системы МДОС1810 выполняются следующие функции: создание и редактирование исходных текстов; трансляция с языков программирования Макроассемблер-86, Бейсик-86, Фортран-86, Паскаль-86; компоновка перемещаемых модулей из отдельно оттранслированных программ и настройка в абсолютный загрузочный модуль; отладка программ; загрузка и выполнение программ; инициализация, форматирование, копирование, восстановление дискет; управление файловой системой.

МДОС1810 обеспечивает работу с дисковыми файлами, динамическое распределение дискового пространства, поддержку директории с древовидной структурой, расширенную обработку ошибок внешних устройств, поддержку средств межмашинного обмена и связи оператора с СМ1810 через консольный терминал в пакетном и диалоговом режимах работы.

В соответствии с планами создания ПО СМ1810 ведется разработка еще двух операционных систем. Это многопользовательская система Микрос-86, являющаяся развитием Микрос-86, и мобильная операционная система ДЕМОС 1810. Все перечисленные операционные системы входят в состав внутреннего ПО СМ1810 и должны поставляться в составе управляющих вычислительных комплексов УВК СМ1810.

Поставка комплексов пользователям. Существуют три

способа поставки заводами УВК СМ1810, предусмотренные в технической документации: в виде типовых комплексов, специфицированных комплексов УВКС и отдельных компонентов или как еще называется такой способ поставки — «россыпью». *Типовой комплекс СМ1810* представляет собой набор модулей, устройств, конструктивов, систем электропитания, комплектов программного обеспечения, документации и ЗИП. Типовой комплекс является поставляемой единицей. В настоящее время разработано восемь типовых комплексов различного состава и конструктивного исполнения — приборный, тумбовый, стоечный и предназначенный для встраивания: УВК СМ1810.10-40, УВК СМ1810.11-41. В процессе развития производства СМ1810 число типовых комплексов будет увеличено.

При способе поставки комплексов СМ1810 в виде УВКС пользователь составляет из имеющегося на заводе-изготовителе, так называемого конфигуратора, определенный набор средств СМ1810, согласовывает его состав с заводом и получает специфицированный комплекс под заданные системные требования. Недостатком способа поставки УВК является увеличение цикла «заказ-поставка», связанного со специальной разработкой общесистемной документации, индивидуального заказа комплектующих изделий и т. д.

Способ заказа отдельных компонентов УВК (модулей, конструктивов, устройств) применяется для комплекта систем, где средства СМ1810 используются для встраивания в оборудование (например, в качестве встраиваемой микроЭВМ в прибор используется модуль МЦП-16) или для расширения функций существующих УВК. Все три способа поставки обеспечивают гибкость построения и эксплуатации систем, построенных на базе СМ1810.

Общее описание технических и программных средств СМ1810, способы их поставки дают некоторое представление об особенностях построения этой микроЭВМ, путях повышения надежности, расширения функциональных возможностей и обеспечения совместимости средств в рамках семейства СМ1800. Открытая магистрально-модульная структура с единым внутренним интерфейсом И41 на существующем этапе развития микропроцессорной техники представляет ряд известных преимуществ, реализованных в СМ1810. С другой стороны, «узким местом» такой структуры является внутренний интерфейс, который в определенной степени ограничивает общую производительность и живучесть всей системы. Возможность организации межпроцессорной шины связи между модулями МЦП-16 с помощью интерфейса ИРПП-М или стыка С2 позволяет резервировать И41. Это повышает

живучесть системы в целом. Функционально-направленное построение основных компонентов структуры модулей по принципу «модуль-функция» упрощает построение всей структуры и привлекает к созданию систем пользователей, не являющихся профессиональными программистами.

В дополнение к традиционным способам развития функциональных возможностей в СМ1810 разработан блок расширения интерфейса И41, позволяющий включать в него процессорные модули, модули системного ОЗУ и модули, работающие с прямым доступом в системную память. Многопроцессорные режимы работы в совокупности со страничной памятью объемом до 16М байт и другими системными особенностями СМ1810 позволяют решать задачи, которые традиционно выполняются большими ЭВМ вычислительных центров. Двухходовые ОЗУ в модулях МЦП-16 и МЦП-1 улучшают динамику функционирования этих модулей и соответственно всей системы в целом. Наличие различных внешних интерфейсов СМ1810 обеспечивает построение многоуровневой магистрально-модульной структуры, а также распределенных и сетевых структур. Разработка нормативных материалов и стандартов СМ ЭВМ на конструктивы, интерфейсы, программное обеспечение и другие компоненты микроЭВМ обеспечивает совместимость средств СМ1810 в рамках моделей семейства СМ1800.

Еще одна модель, разработанная в составе семейства СМ1800, — микроЭВМ СМ1814, представляющая собой вариант СМ1810 для использования в промышленных производствах с ограниченным доступом обслуживающего персонала. Предназначена в основном для применения в локальных технологических сетях, ГПС и АСУТП. В СМ1814 могут входить все модули, блоки расширения, модульные источники питания и кроссы из состава СМ1810. Периферийные устройства — дисплей, и в отдельных случаях печать с дистанционными интерфейсами связи — ИРПС, стык С2. СМ1814 — это УСО в промышленном исполнении. Виды сигналов аналогового и дискретного ввода-вывода аналогичны СМ1810; дополнительно будут введены в состав СМ1814 блок ввода сигналов низкого уровня, термометров сопровитивлений и терморпар СМ9306 и блок формирования поправки СМ9307 из состава СМ1803. Конструктивная особенность СМ1814 заключается в организации двухконтурной схемы вентиляции: внутренний контур (где располагаются монтажные блоки с функциональными модулями) герметичный, внешний пропускает воздушный поток окружающего воздуха через специальную полость с охлаждаемыми поверхностями, а также во введении специальных амортизаторов, позволяющих расширить

диапазон допустимых амплитуды и частоты вибраций.

СМ1814 — без дисковых накопителей, поэтому все программное обеспечение находится в ППЗУ модуля МЦП-16 и модулях перепрограммируемой памяти МПЗ или загружается по каналам связи в ОЗУ. Функционирование СМ1814 в режиме реального времени поддерживается операционной системой ОС СФП-1810.

Изготовление и поставка УВК СМ1810 и СМ1814 будут производиться несколькими заводами, каждый из которых ориентирован на определенные направления применения УВК.

Одним из последних этапов в многоэтапной последовательности создания систем является разработка пакетов прикладных программ, а также технических средств, обеспечивающих создание систем «под ключ». На базе СМ1810 и СМ1814 заканчивается разработка нескольких систем со сроком окончания в 1986 году. Одной из таких систем является автоматизированное рабочее место для конструкторских и технологических работ в машиностроении и металлообработке Автограф 840.

Автоматизированное рабочее место Автограф 840 — это комплекс технических средств, базового программного обеспечения и пакетов прикладных программ, обеспечивает автоматизацию конструкторских и технологиче-

ских работ. Автограф 840 позволяет в интерактивном режиме автоматизировать разработку: конструкций изделий машиностроения, их отдельных узлов и деталей; технологии; изготовления программ для станков с ЧПУ по результатам разработки конструкции и технологии. Технические и программные средства обеспечивают автоматизированную разработку графической, текстовой и технологической документации.

Автоматизированное рабочее место может использоваться автономно либо в составе САПР и автоматизированного конструкторско-технологического бюро. АРМ Автограф 840 включает: специфицированный вычислительный комплекс на базе микроЭВМ СМ1810; устройство отображения графической и алфавитно-цифровой информации Автограф 841; устройство получения твердой копии экрана дисплея Автограф 842; графопостроитель форматов А3 и А0; минипланшет Автограф 843; растровое печатающее устройство СМП 6320; перфокарочную станцию СМ1800.6204.

Программное обеспечение Автограф 840 состоит из базовой интерактивной графической библиотеки Автограф 844, предназначенной для автоматизации разработки прикладных программ, и пакета прикладных программ для разработки конструкторской документации Автограф 845, предназначенного для повышения про-

изводительности труда при выполнении графических работ по созданию чертежей в области механики и металлообработки.

Разработка Автограф 840 велась параллельно с разработкой технических средств СМ1810 и операционной системы БОС1810. В результате изменен традиционный подход к последовательному выполнению этапов разработки средств и сокращены общие сроки создания системы.

Телефон для справок: 455-57-61
(г. Москва)

ЛИТЕРАТУРА

1. Шкамарда А. Н. Метод описания и оценка структур микроЭВМ функционально-модульной архитектуры // Управляющие системы и машины. — 1986. — № 5.
2. Гиглавый А. В., Кабанов Н. Д., Прохоров Н. Л., Шкамарда А. Н. МикроЭВМ СМ1800. Архитектура, программирование, применение. — М.: Финансы и статистика, 1984.
3. Крамфус И. Р., Новик А. Г., Перцов Е. Е. Обзор операционной системы реального времени РМХ/80 // В сб.: Операционные системы реального времени для микроЭВМ. — М.: МЦНТИ, 1984.

Статья поступила 24 июля 1986 г.

УДК 681.326

В. М. Долкарт, В. Н. Степанов, И. Р. Крамфус

ОДНОПЛАТНАЯ МИКРОЭВМ ПМВ02 НА БАЗЕ МПК БИС К1810

Для магистрально-модульных микросредств управляющей вычислительной техники серии В9 (МСУВТ В9), применяемых в промышленных системах автоматизации, была разработана высокопроизводительная 16-разрядная одноплата микроЭВМ типа ПМВ02 на базе МПК БИС серии К1810 [1, 2]. МикроЭВМ (рис. 1) конструктивно выполнена на одной печатной плате площадью 9 дм².

В микросистеме, собираемой из набора плат МСУВТ В7/В9, совместная работа ведущих одноплатных микроЭВМ и ведомых плат ЗУ, устройств ввода-вывода, УСО и т. д. осуществляется через системную межмашинную магистраль (ММ), совместимую с международной системной магистралью IEEE 796. Это позволяет организовывать объединенную симметричную мультипроцессорную вычислительную систему, в которой микропроцессоры с собственной локальной памятью могут связываться через общее разделяемое пространство системной памяти.

Внутренняя локальная магистраль (ВМ) содержит шины адреса, данных и управления и связывает внутрисистемные ресурсы микроЭВМ.

Центральный процессорный узел (ЦПУ) (рис. 2) одноплатной микроЭВМ использует микросхему 16-разрядного микропроцессора (МП) КМ1810ВМ86, генератора тактовых сигналов КР1810ГФ84 и системного контроллера КР1810ВГ88.

Выход ОС генератора КР1810ГФ84 управляет контроллером динамического ОЗУ. Таким обра-

зом, все внутренние ресурсы микроЭВМ работают синхронно с генератором КР1810ГФ84 и внутренний сигнал «Готовность» подается непосредственно на вход генератора.

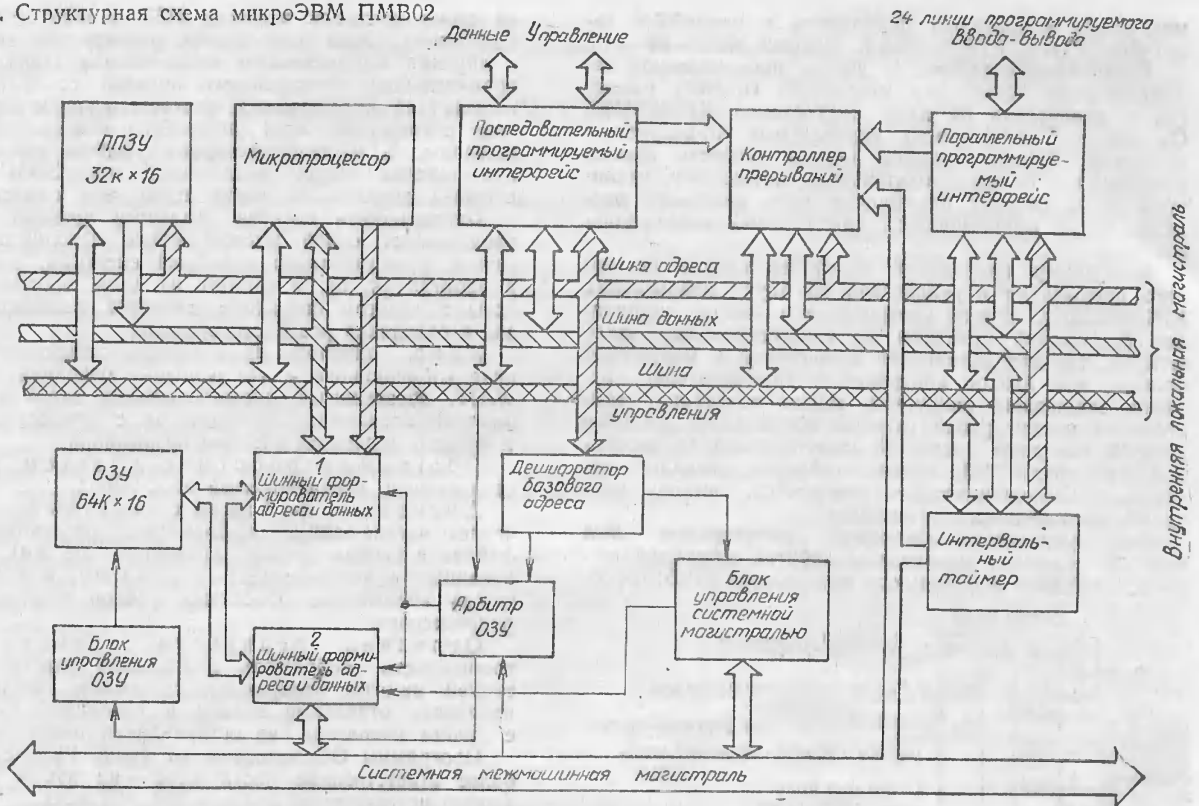
Сигналы «готовности», получаемые из системной магистрали ММ при работе микроЭВМ с ведомыми платами (расширителями памяти и ввода-вывода, контроллерами дисков и т. д.), асинхронны с импульсами генератора КР1810ГФ84.

Отметим, что генератор КР1810ГФ84 имеет внутри лишь один триггер для синхронизации входного сигнала «готовности» с синхронными импульсами МП, что приводит к появлению метастабильного состояния этого триггера и ненадежной работе при формировании сигнала RDY на входе МП, если входной сигнал «готовности» асинхронен по отношению к синхронным импульсам генератора. Чтобы избежать этого, входной сигнал «готовности» принимается в специальный D-триггер, дополнительно синхронизирующий этот сигнал с импульсами генератора и устраняющий появление метастабильного состояния в триггере на входе «готовности».

Системный контроллер КР1810ВГ88 декодирует сигналы состояния МП и вырабатывает сигналы работы с памятью, вводом-выводом и контроллером прерывания, которые выдает на шину управления. Сигнал ALE используется для приема в адресный регистр адреса с мультиплексированных шин адрес-данные МП. Адресный регистр выполнен на трех 8-разрядных буферных регистрах КР580ИР82.

ПЗУ (16К или 64К байт в зависимости от типа устанавливаемых в разъемы микросхем — К573РФ2 или К573РФ4) включает в себя специальную микросхему, содержащую контрольные разряды каждого байта, дополняя число единиц байта до нечетного. Все обращения к ПЗУ контролируются по четности.

Рис. 1. Структурная схема микроЭВМ ПМВ02



Триггер сбоя ПЗУ имеет сигнализацию и может быть подсоединен к контроллеру прерывания своей или другой одноплатной микроЭВМ, используемой в микросистеме.

В ПЗУ содержатся постоянные программы: пользователя, монитор и тесты платы.

Для проверки работы контрольного оборудования один из байтов ПЗУ имеет неправильную четность. Обращение к этой ячейке должно вызвать установку триггера сбоя ПЗУ.

ОЗУ (128К байт) реализовано на микросхемах динамической памяти К566РУ5. Каждый байт имеет контрольный разряд. При записи в него идет информация, дополняющая число единиц до нечетного. При чтении контролируется четность. Триггер сбоя ОЗУ имеет сигнализацию и может быть подсоединен к

контроллеру прерывания своей или другой микроЭВМ. После включения питания программа монитор производит «очистку» ОЗУ, устанавливая таким образом правильные значения контрольных разрядов.

Для проверки работы контрольного оборудования имеется триггер (управляемый как порт ввода-вывода), блокирующий запись информации в контрольный разряд. Таким образом, в ОЗУ можно записать «четную» информацию и при чтении ее вызвать установку триггера сбоя ОЗУ.

ОЗУ — двухпортовое. Один порт обслуживает обращения от внутреннего МП, другой — со стороны системной магистрали ММ.

Шинные формирователи адреса и данных 1 (на базе КР1810ВА86) и 2 (на базе КР1810ВА87) подключают на вход ОЗУ шину адреса и данных соответственно из магистралей ВМ или ММ.

Арбитр ОЗУ решает конфликты при одновременном обращении к ОЗУ со стороны магистралей ВМ и ММ, пропуская обращения только одного порта и включая соответствующие шинные формирователи адреса и данных. Вход со стороны внутрисплатной магистрали ВМ — приоритетный.

Дешифратор базового адреса определяет расположение ОЗУ (128К байт) в пространстве памяти (1М байт) со стороны ВМ и ММ и реализован на «прожигаемом» ПЗУ типа К556РТ5. Кроме того, он по старшим разрядам адресного регистра определяет при каждом обращении к памяти или вводу-выводу, находится ли данная ячейка или порт на плате микроЭВМ или на других платах.

Параллельный интерфейс имеет 24 линии программируемого ввода-вывода и реализован на КР580ВВ55А. Он позволяет передавать в микроЭВМ дискретные сигналы объекта управления и подключать к микроЭВМ графопостроитель, печатающее устройство, программаторы ПЗУ и ПЛМ.

Последовательный интерфейс, реализованный на КР580ВВ51А, обеспечивает межрейтовые связи между

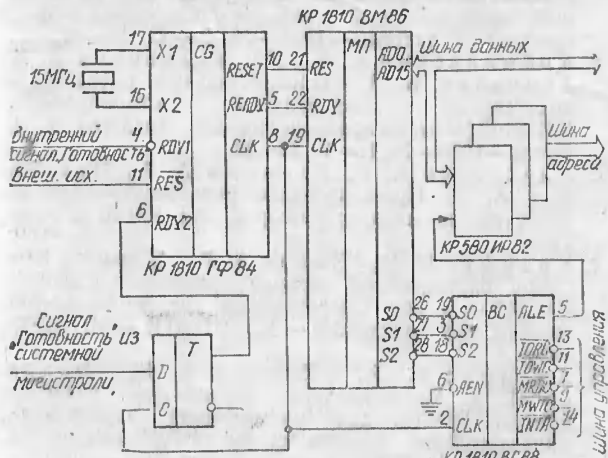


Рис. 2. Блок-схема центрального процессорного устройства

микроЭВМ и позволяет подключать к микроЭВМ теле-тайпы РТА-6, РТА-7, РТА-8, дисплей 15ИЭ—00—013.

Интервальный таймер (с учетом промышленной направленности применения микроЭВМ ПМВ02) расширен и реализован на двух микросхемах КР580ВИ53. Он имеет 6 программно управляемых 16-разрядных счетчиков. Каналы таймера задают скорость обмена информацией по последовательному интерфейсу, реализуют через систему прерывания часы реального времени, могут использоваться для целевых задач пользователя.

Контроллер прерываний имеет 15 входов векторного прерывания и реализован на двух микросхемах КР1810ВН59А. Входы контроллера и выходы источников прерываний выведены на коммутационное поле штырей, так как источников прерывания в микроЭВМ больше, чем входов контроллера. Пользователь, изменяя соединения накруткой, может создавать собственные конфигурации системы прерываний. На поле штырей выведены также 15 шин прерывания системной магистрали ММ, выходы таймеров, сигналы готовности последовательного интерфейса, выходы порта «С» параллельного интерфейса.

Блок управления системной магистралью ММ (рис. 3) содержит микросхемы арбитра системной шины КР1810ВВ89 и системного контроллера КР1810ВГ88.

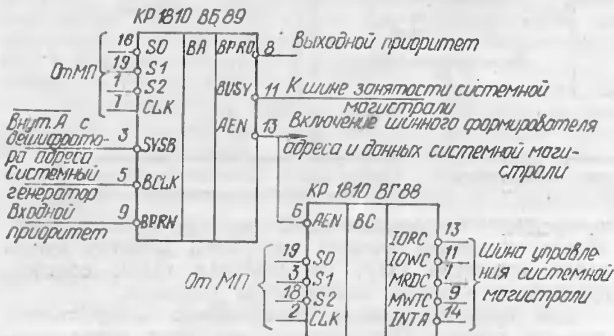


Рис. 3. Блок управления системной магистралью

Сигнал ВНУТ А на выходе дешифратора базового адреса означает, что ячейки памяти или порта ввода-вывода нет на данной плате и обращение к ним должно производиться через ММ. В этом случае микросхема арбитра магистральной КР1810ВВ89 совместно с такими же арбитрами на других платах определяет момент времени, когда владение магистралью ММ передается данной плате. При этом вырабатывается сигнал на выходе АЕН, который подключает к ММ шинные формирователи адреса и данных и включает системный контроллер КР1810ВГ88, вырабатывающий сигнал управления на шину управления ММ.

Системный контроллер обеспечивает гарантированную задержку 110 нс между сигналом АЕН и сигналом на шине управления ММ для того, чтобы предварительно установился адрес на адресной шине ММ.

Программное обеспечение микроЭВМ. Тестовая программа предназначена для налаточных работ и приемосдаточных испытаний микроЭВМ. Она работает в интерактивном режиме, используя дисплей или теле-тайпы, присоединяемые к последовательному интерфейсу микроЭВМ.

Тестовые проверки (по функциональному принципу) охватывают основные узлы микроЭВМ. ПЗУ проверяется контрольным суммированием, ОЗУ — тестом чтения-сравнения-записи с учетом контроля по четности, а также тестом двухпортовости (при этом необходима вторая аналогичная микроЭВМ). Отдельно проверяет-

ся схема контроля четности ПЗУ и ОЗУ («контроль контроля»). Узел прерывания проверяется совместно с другими микросхемами ввода-вывода (таймерами и параллельным интерфейсом) подачей стимулирующих воздействий и получением соответствующих реакций в виде прерываний. Узел управления межмашинной магистралью и многопроцессорная работа проверяются при наличии второй аналогичной микроЭВМ и плат внешней оперативной и/или постоянной памяти.

Операционная система реального времени РМС/86 представляет собой многозадачный исполнитель программ, функционально подобный системам, описанным в работах [4, 5], и состоит из ядра, программы работы с пультом оператора, дисковой файловой системы и отладчика реального времени.

Ядро, опираясь на механизм синхронизации в виде «обменников» и две основные операции SEND и WAIT, обеспечивает диспетчеризацию задач по дисциплине абсолютных приоритетов с дообслуживанием и функционирование системы прерываний.

Программа работы с пультом реализует построчно редактируемый ввод-вывод.

Дисковая файловая система обеспечивает чтение-запись информации последовательных файлов с гибких дисков (диаметром 203 мм), организованных в формате ISIS-II (RMX/80), а также загрузку абсолютных объектных файлов программ микропроцессора.

Отладчик реального времени дает возможность отображать и изменять участки оперативной памяти, отображать системные списки, останавливать отдельные задачи и исполнять процедуры с пульта оператора, не останавливая работу системы.

Программы ОС написаны на языке PL/M-86, а наиболее ответственные части ядра — на языке ASM86 (время переключения задач — 300...400 мкс).

Одноплатная микроЭВМ ПМВ02 вместе с ПО и платами расширения (ОЗУ и ПЗУ, ввода-вывода, УСО и т. д.) после опытной эксплуатации (в течение двух лет) производится серийно.

Опыт эксплуатации микроЭВМ и больших программных систем (объемом до 500К байт) на ее основе еще раз показал настоятельную необходимость применения для отладки программ и аппаратуры микроЭВМ систем разработки с внутрисхемными эмуляторами.

Тел. для справок — 924-72-07 — Степанов Виктор Николаевич (среда с 8⁰⁰ до 16⁰⁰).

ЛИТЕРАТУРА

1. Шереметьевский Н. Н., Долкарт В. М. Магистрально-модульные микросредства управляющей вычислительной техники // Микропроцессорные средства и системы.—1984.—№ 2.—С. 24—27.
2. Кобылинский А. В., Москалевский А. И., Темченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор КМ1810ВМ86 // Микропроцессорные средства и системы.—1986.—№ 1.—С. 28—33.
3. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. М.: Радио и связь, 1984.
4. Крамфус И. Р., Новик А. Г., Перцов Е. Е. Обзор операционной системы реального времени RMX/80.—В кн.: Операционные системы реального времени для микроЭВМ.—М.: МЦНТИ — МНИИПУ, 1984.
5. Крамфус И. Р., Перцов Е. Е., Роднова И. А. Многозадачный исполнитель реального времени РМС/80 для микросредств управляющей вычислительной техники серии В7 // Электротехника.—1982.—№ 5.

Статья поступила 20 июня 1986 г.

В. Н. Говорун, Н. В. Горбунов, П. В. Мамаков, В. Г. Рыбаков, А. Н. Сытин, Г. М. Холоденко

МИКРОЭВМ И УПРАВЛЯЮЩИЕ МОДУЛИ НА БАЗЕ МПК БИС СЕРИИ K1810

В Институте физики высоких энергий разработаны и широко применяются модули в системе СУММА на основе 8-разрядного МПК БИС серии K580 [1].

Характеристики МПК БИС серии K1810 [2] — высокое быстродействие, мощная система команд, возможность прямой адресации памяти емкостью до 1М байт и др. — позволяет увеличить производительность систем с использованием БИС КМ1810ВМ86 практически на порядок по сравнению с системами на базе БИС КР580ИК80 [3]. На базе МПК БИС K1810 [4] разработаны дополнительный контроллер канала каркаса и драйвер ветви системы СУММА (КАМАК) и микроЭВМ.

Дополнительный контроллер каркаса связывает микроЭВМ с каналом каркаса в соответствии с интерфейсом КАМАК [5]. В управляющем режиме контроллер управляет операциями канала каркаса, обеспечивая обмен данными между микроЭВМ и модулями в каркасе. В подчиненном режиме контроллер выполняет команды каркаса, вырабатываемые другим контроллером. Подчиненный режим обеспечивает доступ к памяти микроЭВМ через канал каркаса.

Драйвер ветви управляет каркасными контроллерами, объединенными магистралью ветви [5], и может работать под управлением микроЭВМ или канала каркаса (род управления выбирается с помощью переключки на плате драйвера). Каждое устройство размещается на одной печатной плате. Объединяя плату микроЭВМ с платой дополнительного контроллера, можно сделать интеллектуальный дополнительный контроллер каркаса (рис. 1, а); можно объединить плату микроЭВМ с платой драйвера ветви в интеллектуальный драйвер ветви (рис. 1, б).

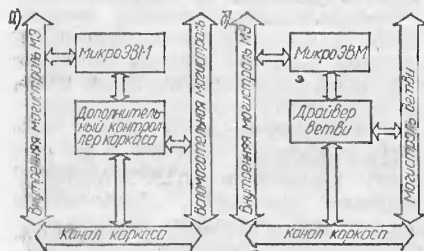


Рис. 1. Схемы объединения отдельных плат для построения интеллектуального дополнительного контроллера каркаса (а) и интеллектуального драйвера ветви (б)

Блок-схема микроЭВМ (рис. 2) включает в себя центральный процессор (ЦП), ОЗУ и ППЗУ, два последовательных интерфейса, контроллер прерываний, контроллер прямого доступа в память (ПДП) и регистр старших адресов ПДП, дешифратор адресов внешних устройств. Эти устройства объединяются внутренней магистралью, которая (через многоконтактный разъем XI на передней панели модуля) служит и для подключения к микроЭВМ дополнительной памяти и внешних устройств.

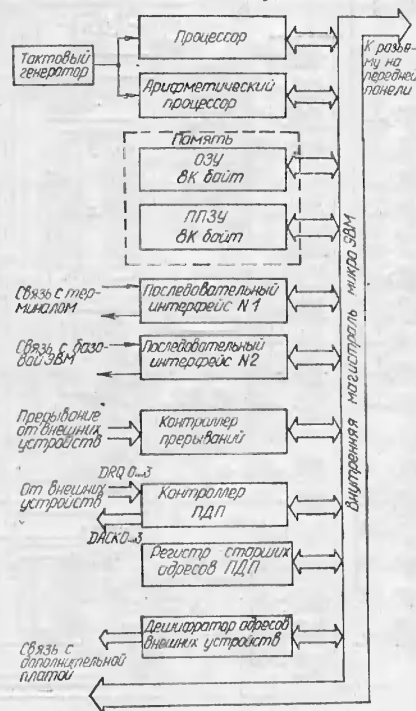


Рис. 2. Блок-схема микроЭВМ

Центральный процессор (рис. 3) построен на базе БИС КМ1810ВМ86 [2]. Его характеристиками являются: архитектура, разработанная для мощного языка программирования ассемблера и эффективная для языков высокого уровня; 14 регистров (16-битных) с симметричными операциями; 24 моды адресации операнда; операции с битами, байтами, словами и блоками; арифметические операции с 8- и 16-битными данными со знаком и без знака в двоичной или десятичной форме, включая умножение и деление; интерфейс, совместимый с системой MULTIBUS.

Центральный процессор включает в себя системный контроллер КР1810ВГ88, адресные регистры на микросхемах КР580ИР82, обеспечивающие разделение адреса и данных от мультилексированной шины адреса-данных процессорного элемента, и шинные формирователи на микросхемах КР580ВА86. Возможности микроЭВМ расширить просто, подключив дополнительный арифметический процессор К1810ВМ87 [4].

Генератор тактовых сигналов (5 МГц) для процессора построен на микросхеме КР580ГФ24 с синхронизирующей кварцевым резонатором на частоту 14745,6 кГц.

Для ОЗУ (8К байт) используются микросхемы К132РУ7 с временем доступа к памяти 250 нс, на скорости работы процессора это не скажется.

ППЗУ (8К байт) выполнено на микросхемах К573РФ2 с организацией 2К×8. Время доступа к памяти названных микросхем — 450 нс. При работе с данной памятью вводится один такт ожидания в цикл операции с использованием сигнала готовности (READY) процессора. Емкость ППЗУ можно увеличить, установив в гнезда микросхемы К573РФ4 (8К×8) вместо микросхем К573РФ2, с изменением коммутации цепей выбора отдельных микросхем.

Два последовательных интерфейса (на микросхемах КР580ВВ51А) работают в асинхронном режиме на 4-проводную линию. Приемники (с оптронным переключателем К293ЛП1) и передатчики (с транзисторным оптроном АОТ110) интерфейсов могут работать в пассивном и активном режимах (выбирается переключками), обеспечивая работу на токовую петлю 20 мА. Скорости передачи (75...9600 бод) задаются переключками отдельно для каждого интерфейса.

Интерфейсы используются для связи с терминалом и с базовой ЭВМ. В последнем случае обмен данными возможен под управлением программы и по прерываниям.

Контроллер прерываний (КР580ВН59А) — позволяет принимать, маскировать и приоритетно кодировать запросы от восьми источников прерываний. Выбор сигналов прерываний и присваивание им приоритетов производится переключками на плате.

Три источника прерываний находятся внутри микроЭВМ: сигнал от кнопки (или высокочастотного разьема) на передней панели, от последовательного интерфейса «МикроЭВМ—базовая ЭВМ» и от дополнительного процессора (при обнаружении им ошибки в вычислениях). В качестве других сигналов прерываний можно использовать сигналы от плат дополнительного контроллера, драйвера ветви, а также от многоконтактного разъема на передней панели модуля.

Сигналы от источников прерываний записываются в контроллер по их репедам, где они маскируются, приоритетно кодируются, вырабатывают сигнал прерывания для процессора. После подтверждения прерывания от процессора вырабатывается инструкция вызова подпрограммы по адресу, предварительно записанному в контроллер при его программировании.

Контроллер ПДП (КР580ИК57) — обеспечивает обмен данными между одним из четырех портов ввода-вывода и ОЗУ без участия ЦП. При предварительном программировании ПДП указывается номер канала внешнего устройства, начальный адрес памяти и длина массива данных. Взаимодействие с внешним устройством происходит по принципу «Запрос прямого доступа» от внешнего устройства (DRQ) — «Подтверждение прямого доступа» (DACK) внешнему устройству. Работа ЦП блокируется сигналом удержания, поступающим от контроллера на вход Hold ЦП.

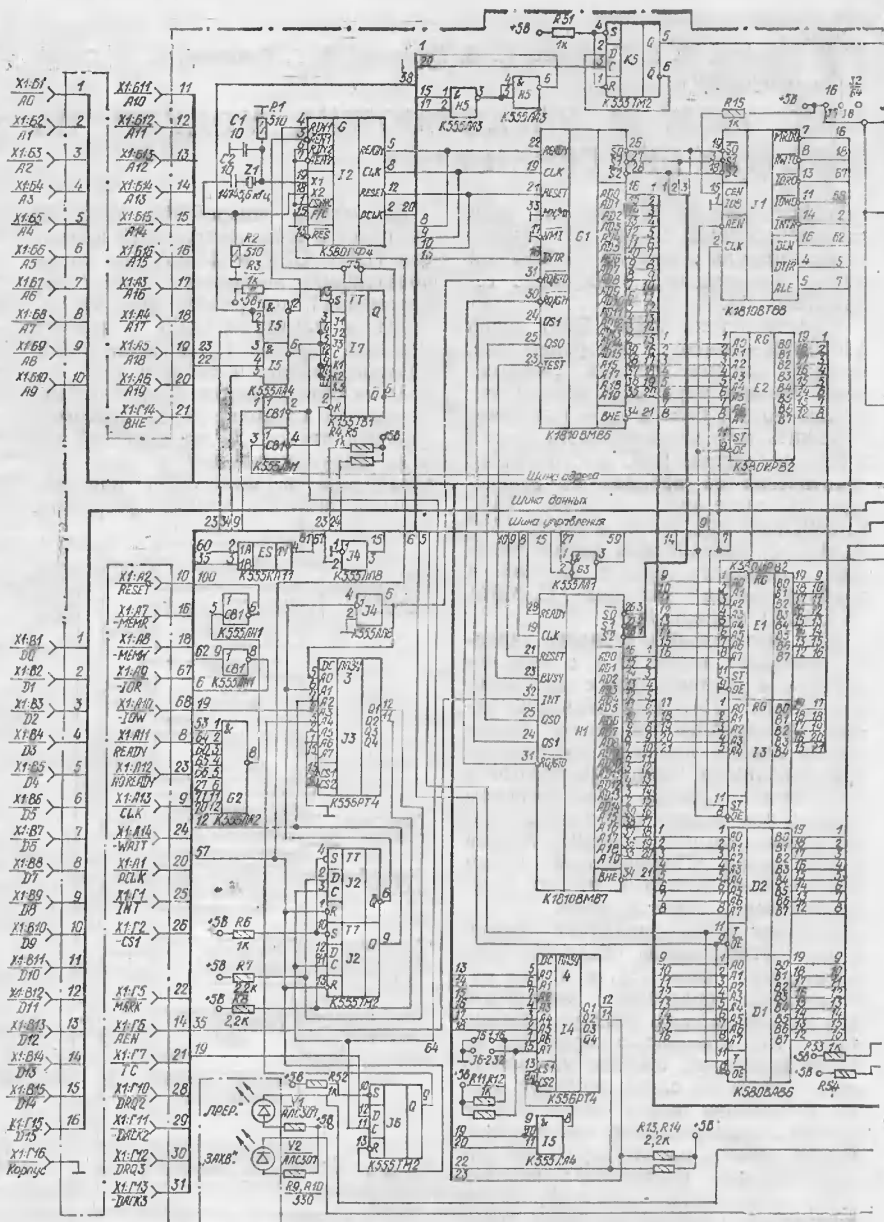
Два канала ПДП (№ 0 и № 1) служат для связи с регистрами ввода-вывода дополнительного контроллера в управляющем и подчиненном режимах. Для связи драйвера ветви с регистрами данных используется один из этих каналов. При работе каналов происходит обмен 16-битными словами. Контроллер ПДП управляет 16 адресными линиями, в то время как внутренняя магистраль имеет 20 линий. Поэтому четыре старших бита адреса задаются с помощью регистра старших адресов (загружаются предварительно). Два других канала (№ 2 и № 3) служат для обмена байтами и расширения системы внешними устройствами (например, ИГМД). Цепи их сигналов («Запрос на прямой доступ» и «Подтверждение прямого доступа») выводятся на многоконтактный разъем на передней панели.

Дешифратор адресов внешних устройств позволяет выбирать устройства на плате микроЭВМ и на других платах.

МикроЭВМ снабжена схемой генерации строб-импульса, вырабатываемого при выборе одного из внешних устройств и подключенного к высокочастотному разъему на передней панели. Этот сигнал может служить для синхронизации внешних приборов с работой микроЭВМ (например, для запуска развертки осциллографа).

Кроме ППЗУ типа К573РФ2, содержащего программу монитора и пользовательские программы, на плате есть еще пять ППЗУ типа К556РТ4, выполняющих следующие функции:

- ППЗУ 1 — дешифратор адресов RAM, находящихся на плате;
- ППЗУ 2 — дешифратор внешних устройств;
- ППЗУ 3 — осуществляет протокол RQ/GT [3] работы центрального процессора;



ППЗУ 4 — дешифратор адресов EPROM, находящихся на плате;

ППЗУ 5 — схема управления работой КРДП (осуществляет байтовый обмен по каналам ПДП № 2 и № 3, и обмен 16-битными словами по каналам № 0 и № 1).

В качестве расширения конфигурации микроЭВМ в настоящее время разработаны модуль динамической памяти (0,5М байт) и контроллер накопителей на гибких магнитных дисках. Оба модуля выполнены также в конструктиве СУММА (КА-МАК).

МикроЭВМ не имеет никакой связи с каналом каркаса, а с дополнительным контроллером или драйвером

ветви должна связываться специальным жгутом или с помощью дополнительных разъемов на платах.

Программное обеспечение микроЭВМ содержит модифицированный монитор SDK-86 системы MSC-86 [6], кросс-ассемблер на ЭВМ DES-10 и CM-4 [7] и дисковую операционную систему CP/M-86.

Программа монитор (4К байт) хранится в ППЗУ. Монитор запускает программы, обрабатывает прерывания, позволяет производить чтение-запись содержимого памяти, регистров процессора и регистров внешних устройств, задавать точки останова программ пользователя. В монитор входит также программа-загрузчик рабочих программ из базовой ЭВМ. Обработчик прерыва-

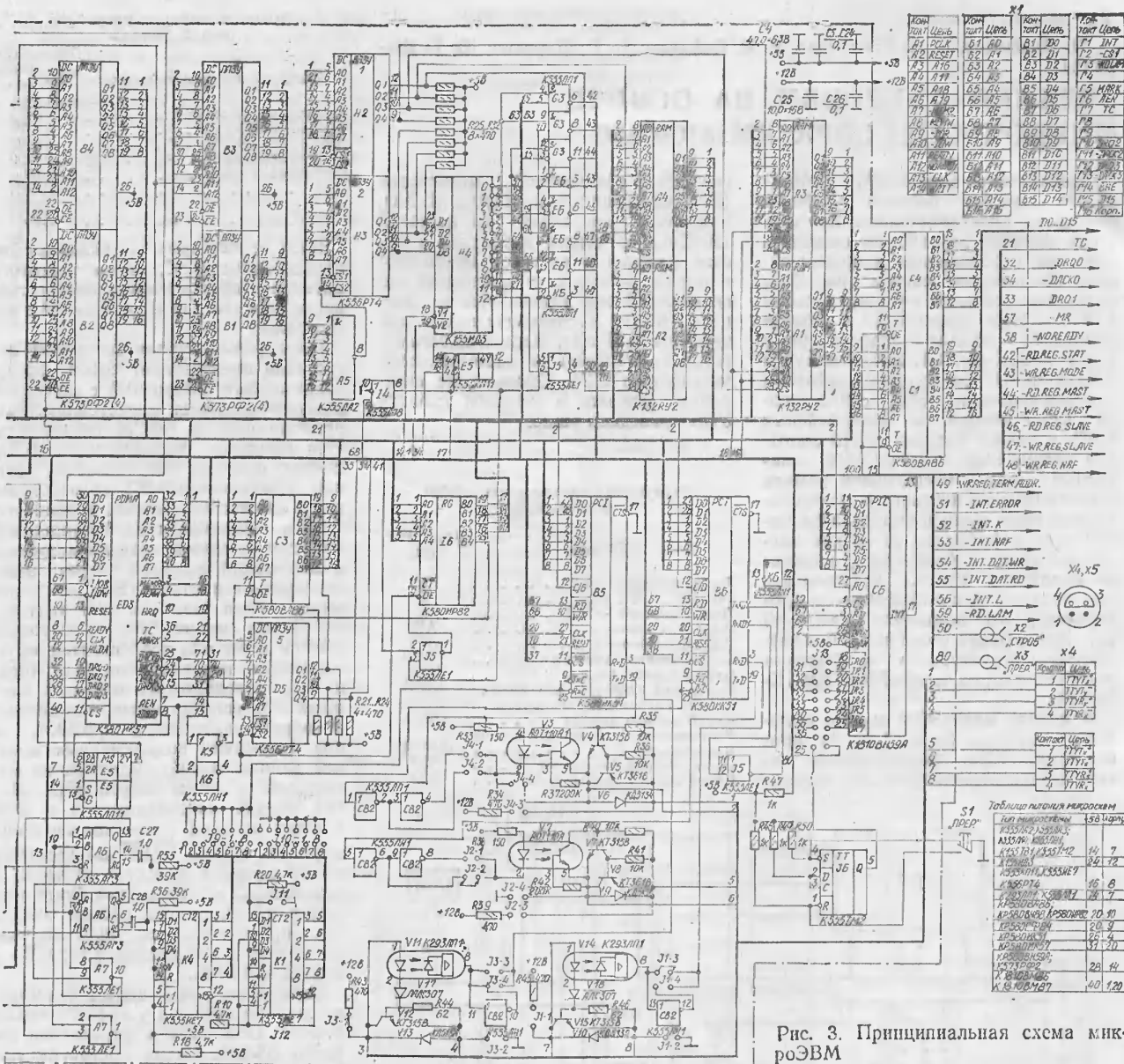


Рис. 3. Принципиальная схема микроЭВМ

ний монитора сохраняет содержимое регистров, статус процессора и передает управление на обслуживающую программу (в ОЗУ) пользователя.

Устройства внедряются в системах съема информации автоматизированных экспериментальных установок, а также в системах управления электрофизическими установками ИФВЭ [8].

Адрес для запросов о приобретении технической документации: 142200, г. Серпухов, Институт физики высоких энергий.

ЛИТЕРАТУРА

1. Говорун В. Н., Ермолин Ю. В., Мамаков П. В. и др.

Микропроцессорные модули для автоматизированных установок физики высоких энергий. // Микропроцессорные средства и системы.— 1985.— № 1.

2. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и миниЭВМ.— М.: Энергоатомиздат, 1984.

3. Intel Component Data Catalog.— Santa Clara, Calif., Intel Corp., 1981.

4. The 8086 Family User's Manual. Numeric Supplement.— Santa Clara, Calif., Intel Corp., 1980.

5. CAMAC. Updated Specifications.— EUR 6500e, ECA/ESONE Committee, 1983, vol. 1.

6. MSC-86 User's Manual.— Santa Clara, Calif., Intel Corp., 1979.

7. Ковальцов В. И., Максимов Г. М. Кросс-обеспечение для 16-разрядного микропроцессора.— Серпухов: ИФВЭ, 1985, препринт 85—96.

8. Говорун В. Н., Горбунов Н. В., Ермолин Ю. В. и др. МикроЭВМ и управляющие модули на базе микропроцессорной серии К1810.— Серпухов: ИФВЭ, 1985, препринт 86—16.

Статья поступила 27 июня 1986 г.

УЧЕБНАЯ МИКРОЭВМ НА ОСНОВЕ МИКРОПРОЦЕССОРА КМ1810ВМ86

Одноплатная микроЭВМ УМПК-86/ВМ является основой учебного комплекса, предназначенного для изучения методов разработки, схемотехнических особенностей и программного обеспечения систем на базе микропроцессорного комплекта серии К1810. Кроме микроЭВМ в состав комплекса входят лабораторный стенд УМПК-86/ЛС1, набор модулей УМПК-86/МИ для изучения работы и программирования БИС сопроцессора ввода-вывода и интерфейсных БИС, а также модули УМПК-86/МР для расширения возможностей микроЭВМ: модуль оперативной памяти с контроллером цветного графического дисплея и модуль контроллера накопителя информации на гибких магнитных дисках. Кроме того, в составе комплекса можно использовать ряд модулей из комплектации микропроцессорного комплекса УМПК-80 [1]. Ниже будут более подробно рассмотрены структура и особенности режимов работы микроЭВМ УМПК-86/ВМ.

В модуле микроЭВМ можно выделить следующие блоки (рис. 1): процессоров (БП), формирования системной магистрали (БФСМ), формирования резидентной магистрали (БФРМ), дешифрации адреса (БДА), индикации состояния магистралей (БИСМ), а также входящие в состав локальных ресурсов микроЭВМ и подключенные к ее локальной магистрали блоки клавиатуры и дисплея (БКД), последовательного интерфейса (БПСИ), параллельного интерфейса (БПРИ) и запоминающих устройств (БЗУ). Обозначение внутренних сигналов и сигналов выборки устройств ВВВ приведено ниже.

рования резидентной магистрали (БФРМ), дешифрации адреса (БДА), индикации состояния магистралей (БИСМ), а также входящие в состав локальных ресурсов микроЭВМ и подключенные к ее локальной магистрали блоки клавиатуры и дисплея (БКД), последовательного интерфейса (БПСИ), параллельного интерфейса (БПРИ) и запоминающих устройств (БЗУ). Обозначение внутренних сигналов и сигналов выборки устройств ВВВ приведено ниже.

Внутренние сигналы микроЭВМ УМПК-86/ВМ

| Сигнал | Обозначение |
|--|-------------|
| Линия питания +5 В | А |
| Общий | В |
| Линия питания +12 В | С |
| Адресная линия 0 | АРО |
| Адресная линия 1 | АР1 |
| Строб чтения данных из УВВ | ЧВВ |
| Строб записи данных в УВВ | ЗВВ |
| Состояние клавиши, шаг цикла | КШЦ |
| Работа системной магистрали | РСМ |
| Строб записи адреса | СТБА |
| Разрешение работы системной магистрали | РРСМ |
| Строб чтения данных из ЗУ | ЧЗУ |
| Строб записи данных в ЗУ | ЗЗУ |

Сигналы выборки устройств ввода-вывода

| Выбираемое устройство | Обозначение сигнала выборки |
|-----------------------|-----------------------------|
| Регистр управления | УПР |
| БИС КР580ВВ79 | ВК79 |
| БИС КР580ВВ51А | ВК51 |
| БИС КР580ВВ53 | ВК53 |
| БИС КР580ВВ55А | ВК55 |
| ОЗУ | ОЗУС, ОЗУМ |
| ПЗУ | ПЗУ |

Блок процессоров БП включает две процессорные БИС, схемы тактового питания и обработки сигналов готовности резидентной и системной магистралей.

Блок формирования системной магистрали представляет собой схему, сопрягающую микроЭВМ с системной многопроцессорной магистралью. Разрешение работы БФСМ осуществляется сигналом, формируемым БДА. В свою очередь, БФСМ выдает сигнал, указывающий БП на необходимость обработки сигнала готовности.

Блок формирования резидентной магистрали служит для подключения к микроЭВМ модулей через резидентную магистраль (РМ), а также выдачи ряда сигналов для внутренних узлов микроЭВМ. Построение БФРМ имеет ряд особенностей, обусловленных неопределенностью числа и типа используемых модулей. Сигналы РМ формируются независимо от обращения к ней микроЭВМ, но при отсутствии подключенных модулей данные только выдаются на магистраль, а прием запрещается. Любой модуль, подсоединенный к РМ, в ответ на свой адрес должен сформировать сигнал RSEL, по которому обмен данными между локальной и резидентной магистралями будет разрешен, но запрещена работа БФСМ. Этот прием позволяет упростить схему за счет подключения БЗУ, БДА и БИСМ к БФРМ.

Блок дешифрации адреса формирует сигналы выборки устройств ввода-вывода, ОЗУ, ПЗУ и управления работой БФСМ. Схема построена так, что сигнал разрешения работы системной магистрали выдается в тот момент, когда на локальной и резидентной магистралях нет устройств, распознавших свой адрес, и текущий цикл не является циклом обработки прерывания. Блок ЗУ включает ОЗУ и ПЗУ объемами по 2К и 4К шестнадцатеразрядных слов.

Блок клавиатуры и дисплея служит для подключения к локальной магистрали функциональной клавиатуры и матричного дисплея, используемых при работе с монитором микроЭВМ.

Блок последовательного интерфейса позволяет сопрягать микроЭВМ с периферийными устройствами, осуществляющими обмен информацией в последовательном коде с использованием интерфейса «20 мА токовая петля», и магнитофона.

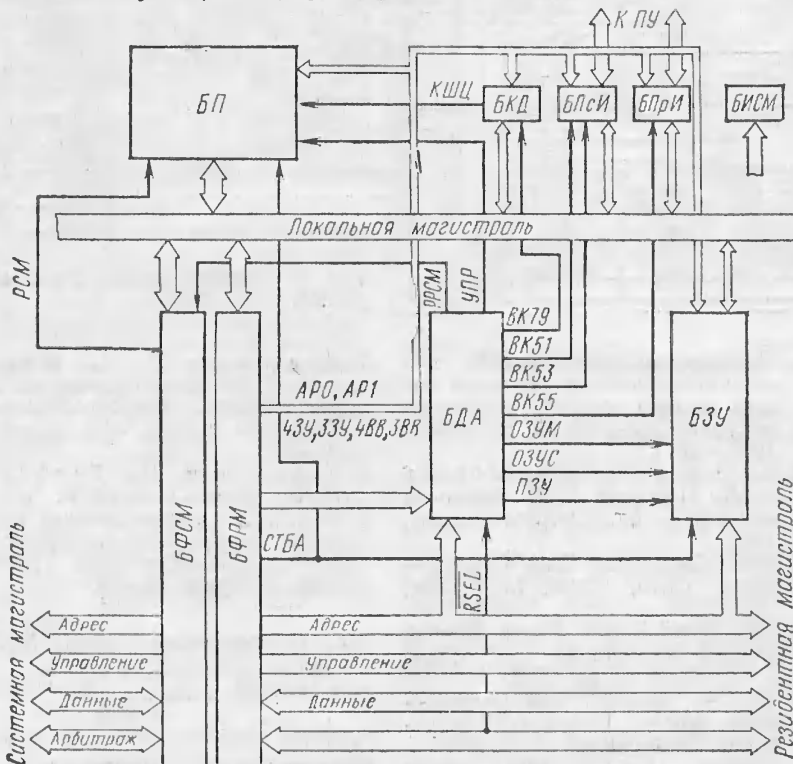


Рис. 1. Структурная схема микроЭВМ УМПК-86/ВМ

Блок параллельного интерфейса позволяет использовать с микроЭВМ ряд устройств, обменивающихся информацией в параллельном коде: печатающее устройство, цифро-аналоговые и аналого-цифровые преобразователи, схемы пользователя и т. д. Блок индикации состояния магистралей дает возможность наблюдать процессы обмена данными между БИС процессоров и устройствами, подключенными к магистралям микроЭВМ.

Принципиальная схема УМПК-86/ВМ реализована на основе МПК серий К1810, К580 и микросхем серий К555 [2]. Для построения ОЗУ и ПЗУ использованы БИС серий К537 и К573 (см. ниже). Модуль имеет четыре разъема, из которых Х1 используется для подключения к системной магистрали, Х2 — к резидентной магистрали, Х3 — к линиям БПСИ, Х4 — к линиям БПри.

Основу блока процессоров (рис. 2) составляют МП БИС D12, работающая в максимальном режиме, и БИС сопроцессора арифметического расширителя D13. Для формирования тактовых сигналов, сигналов начальной установки и готовности используется микросхема D9, входящая в состав МПК. Задание частоты осуществляется кварцевым резонатором BQ1, а подача сигнала начальной установки —

Перечень комплектующих элементов микроЭВМ

| Элемент | Обозначение на принципиальной схеме |
|------------|---|
| К555ЛН1 | D1, D16, D17, D23, D30, D31, D34, D37, D53, D73...D75 |
| К555ТМ2 | D2, D8 |
| К555ЛАЗ | D3, D4, D63, D64 |
| К555ЛЕ1 | D5, D7 |
| К555ТМ8 | D6 |
| КР1810Ф84 | D9 |
| К555ТЛ2 | D10, D5 |
| К555ЛЛ1 | D11, D32, D46, D47 |
| КМ1810ВМ86 | D12 |
| КМ1810ВМ87 | D13 |
| КР580ВА87 | D14, D15, D24, D25 |
| КР580ИР83 | D18...D20, D27...D29 |
| КР1810ВГ88 | D21, D33 |
| КР1810ВБ89 | D22 |
| К555ЛЛ1 | D26, D60, D68, D76 |
| К555ЛАЗ | D38 |
| К155ЛЕ3 | D39, D41 |
| К555ЛАЗ | D40, D42, D43 |
| К155РЕ3 | D44 |
| КР556РТ4 | D45 |
| КР537РУ8А | D48, D49 |
| К573РФ3 | D50 |
| КР580ВВ79 | D51 |
| КР580ВА86 | D51, D70...D72 |
| К554СА3А | D56 |
| К293ЛП1Б | D57 |
| АОТ128А | D58 |
| К555ЛП5 | D59, D61, D62, D67 |
| КР580ВБ51А | D65 |
| КР580ВЛ33 | D66 |

| | |
|-----------------|-----------------------------|
| КР580ВБ55А | D69 |
| К53-4А-6В-22 | мкФ C1, C10 |
| КМ-5Б-П33-16 | пФ C2 |
| КМ-5Б-Н90-0,1 | мкФ C3, C4 |
| КМ-5Б-Н90-0,047 | мкФ C5, C8, C9 |
| КМ-5Б-Н30-3300 | пФ C6, C7 |
| ОМЛТ-0,125-100 | Ом R1, R3, R10, R19...R20 |
| НР1-4-0-15 | кОм R2, R4...R6, R8, R9, E4 |
| НР1-4-9-1 | кОм E1, R7, E2, E3, R30 |
| ОМЛТ-0,125-47 | Ом R11...R18, R42 |
| НР1-4-9-470 | Ом R27, R36 |
| ОМЛТ-0,125-20 | кОм R43...R66, R28, R35 |
| ОМЛТ-0,125-1,6 | кОм R29, R40, R41 |
| ОМЛТ-0,125-160 | кОм R31 |
| ОМЛП-0,125-10 | кОм R32 |
| ОМЛТ-0,125-65 | кОм R33 |
| ОМЛТ-0,125-560 | Ом R34, R37...R39 |
| КД521А | VD1...VD4 |
| КРПМО1А-1К | HL1...HL45 |
| КТ350А | VT1...VT8 |
| КТ315Д | VT9, VT10 |
| КТ361Д | VT11 |
| ВМ1-2 | SA1...SA11 |
| ПКН-150 | SB1...SB28 |
| СНП58-С4 | X1, X2 |
| СНО53-8 | X3 |
| СНО51-40 | X4 |

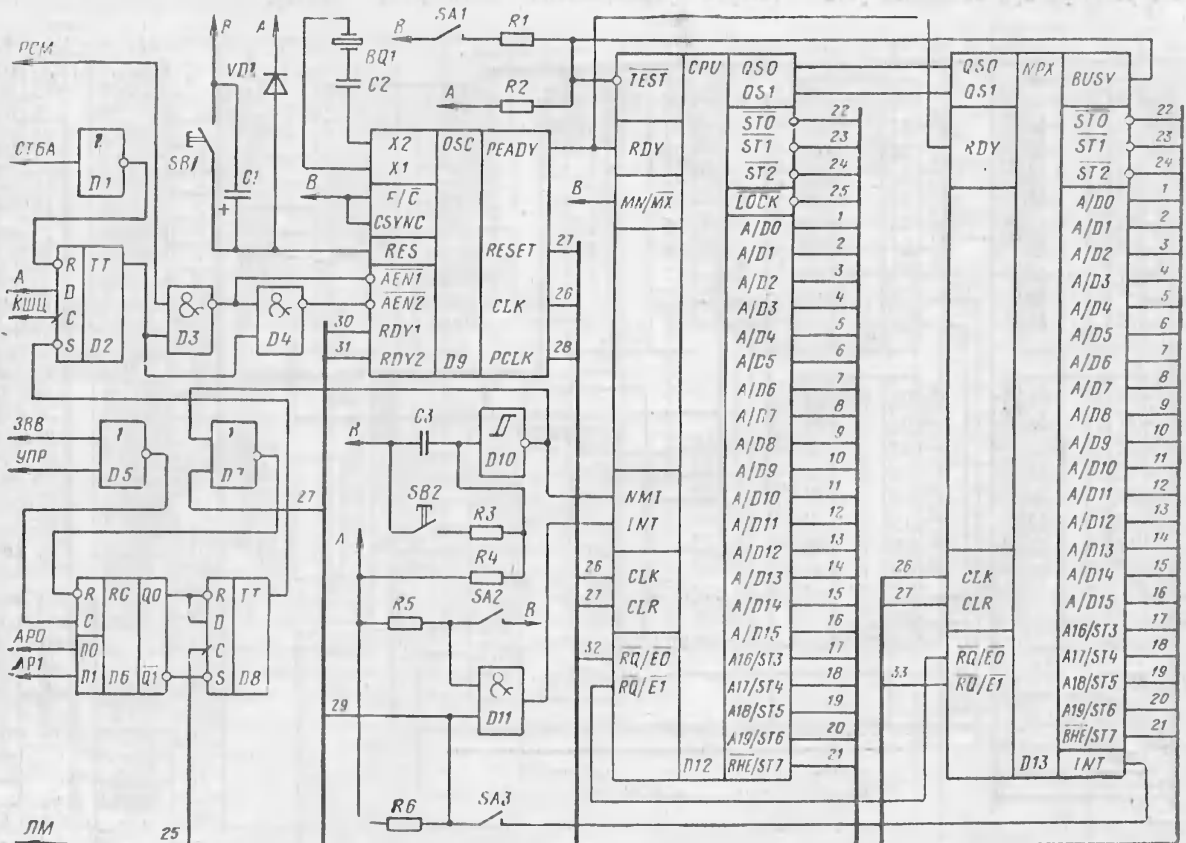


Рис. 2. Принципиальная схема блока процессоров

при включении питания или по требованию пользователя — элементами SB1, C1, VD1. Для формирования запроса немаскируемого прерывания используется клавиша SB2 (Стоп) со схемой устранения дребезга, реализованной на элементах D10, C3, R3, R4. Учитывая, что БИС D13 является не обязательным элементом микроЭВМ и может не устанавливаться, переключатель SA1 дает возможность имитировать формируемый ею сигнал. Коммутация внешнего запроса прерываний выполняется ключом D11 по сигналу от переключателя SA2. При отсутствии внешнего задатчика запроса прерывания и блока приоритетных прерываний пользователь имеет возможность исследовать процесс обслуживания прерываний с использованием вектора 0, формируемого автоматически за счет считывания кода 00 с локальной магистрали. С помощью переключателя SA3 можно также задействовать вектор 0 для подпрограмм обслуживания прерываний сопроцессора.

Схема обработки сигналов готовности реализована на микросхемах D1...D8. Управление режимом выполнения программ осуществляется с помощью регистра управления D6 и триггера D8. Элемент D5 формирует строб записи управляющего кода в регистр, а элемент D7 — сигнал сброса регистра при начальной уста-

новке или нажатии клавиши Стоп. При сбросе регистра или записи в него кода 00 программы выполняются в реальном масштабе времени. Запись кода 01 дает возможность перейти к поциклическому режиму по сигналу LOCK МП БИС, а кода 11 — сразу же после его записи. Использование сигнала LOCK позволяет привязать момент перехода к любой нужной команде. Схема, обеспечивающая выполнение команд по циклам обращения к магистрали, построена на элементах D1...D4. При появлении на выходе триггера D8 низкого уровня сигнала МП БИС переводится в режим ожидания. Текущий цикл можно завершить подачей импульса, формируемого при нажатии клавиши CYCLE. При выполнении следующего цикла процессы повторяются. Схема позволяет также подключать к входам сигнала RDY процессорных БИС сигнал готовности резидентной или системной магистрали. При работе последней формируется сигнал PCM, указывающий на необходимость обработки сигнала готовности. В противном случае воспринимается сигнал с резидентной магистрали.

Блок формирования системной магистрали (рис. 3) состоит из буфера магистрали данных D14, D15, регистров формирования магистрали

адреса D18...D20, системного контроллера D21, формирующего магистраль управления и сигналы управления для ИМС D14...D20, а также арбитра магистрали D22, участвующего в арбитраже доступа к системной магистрали. При получении права доступа к магистрали формируется сигнал PCM, разрешающий работу регистров формирования адреса и системного контроллера, который, в свою очередь, разрешает работу буфера данных. Этот же сигнал используется для разрешения обработки сигнала готовности магистрали. Элементы D16, D17 служат для усиления тактовых сигналов. Переключатель SA4 позволяет подавать на линию тактовых импульсов магистрали сигнал от генератора тактовых импульсов микроЭВМ, а переключатель SA5 — реализовывать различные режимы арбитража магистрали (последовательный, циклический, параллельный).

В системной магистрали приняты следующие сигналы:

ADRI...ADRI9 — адреса, используемые для выбора исполнительного устройства (ячейки ЗУ или УВВ);

ADRO, BHEN — адреса, определяющие к какому байту данных происходит обращение (10 — младший байт, 01 — старший байт, 11 — оба байта);

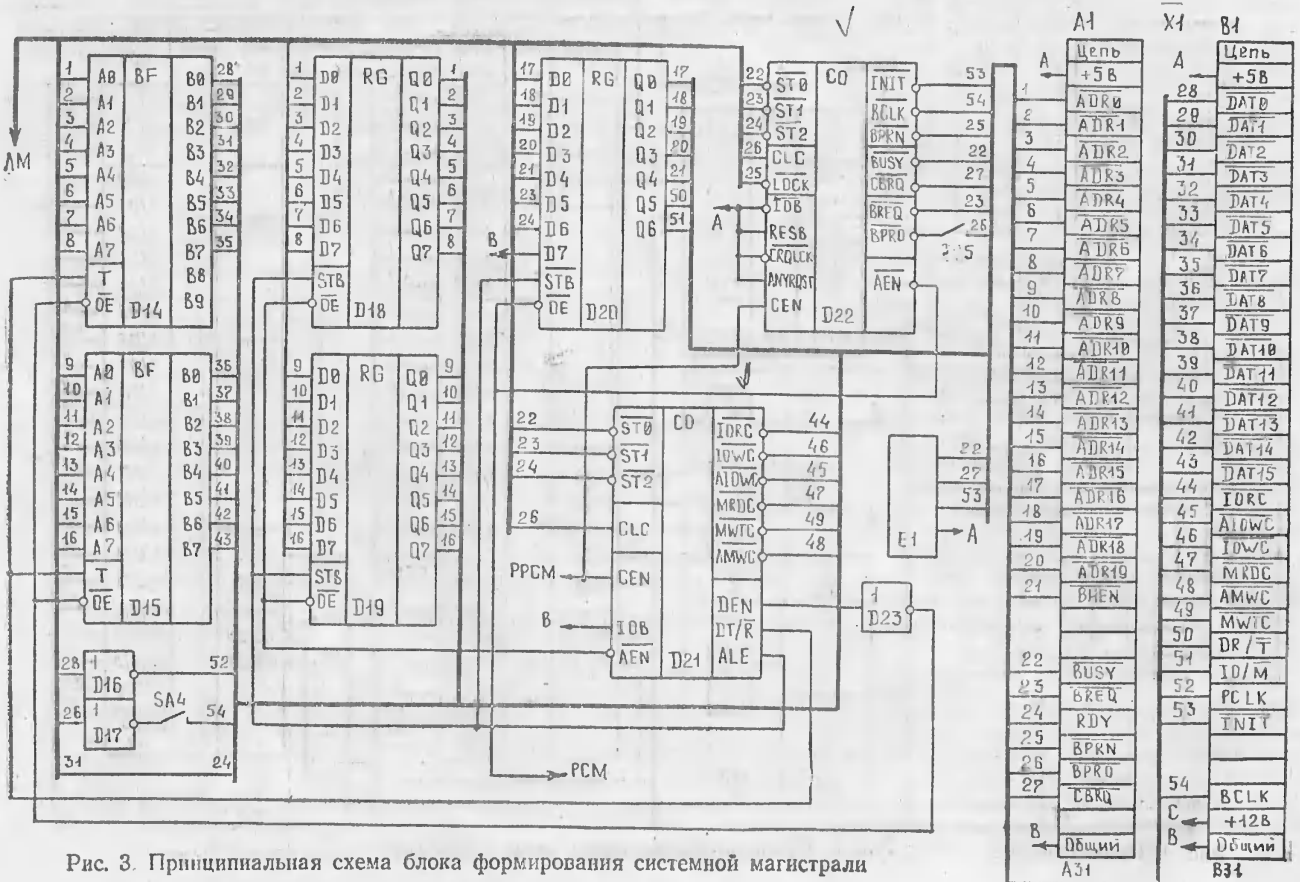


Рис. 3. Принципиальная схема блока формирования системной магистрали

DATA0...DATA15 — двунаправленные линии данных;

BUSY — сигнал занятия магистрали;

BREQ — запрос магистрали (используется при параллельном и циклическом арбитражах);

BPRN — разрешение доступа к магистрали (разрешает арбитру занять магистраль);

BPRO — отказ от доступа к магистрали (используется при последовательном арбитраже);

CBRQ — общий запрос магистрали;

IORC — чтение из УВВ;

IOWC — запись в УВВ;

AIOWC — увеличенный по длительности строб записи в УВВ;

MRDC — чтение ЗУ;

MWTC — запись в ЗУ;

AMWC — увеличенный по длительности строб записи в ЗУ;

DR/T — сигнал, указывающий направление передачи информации по линиям данных (высокий уровень — прием данных в микроЭВМ);

IO/M — сигнал, определяющий вид исполнительного устройства (высокий уровень — УВВ, низкий — память);

PCLK — линия тактовой частоты для модулей;

BCLK — линия синхронизации магистрали;

RDV — линия готовности;

INIT — системный сигнал начальной установки.

К линиям адреса, данных, стробов управления должны подключаться передатчики, имеющие третье состояние, к линиям BUSY, CBRQ, RDV, INIT — с открытым коллектором, а к остальным линиям — с ТТЛ-выходом.

Блок формирования резидентной магистрали (рис. 4) построен аналогично БФСМ. Исключением является отсутствие арбитра доступа, так как микроЭВМ является единственным процессорным устройством на резидентной магистрали. Для осуществления прямого доступа к памяти, подключенной к РМ, необходимо воспользоваться сигналом запроса-разрешения доступа к локальной магистрали RQ/EO или RQ/EI для отключения от нее процессора, а затем сигналом RAEN перевода всех выходов БФРМ в третье состояние. Элементы D30...D32 служат для усиления сигналов, а D34...D36 для инвертирования. Переключатели SA6, SA7 позволяют коммутировать сигналы запроса-разрешения доступа к локальной магистрали. Элемент D26 преобразует прием в микроЭВМ данных при отсутствии на РМ устройств,

распознавших свой адрес и сформировавших ответный сигнал.

В резидентной магистрали используются следующие сигналы:

INT — запрос прерывания, формируемый КПП;

INTA — подтверждение обслуживания запроса прерывания;

RAEN — отключение БФРМ от магистрали;

RESET — начальная установка модулей, подключенных к РМ;

RSEN — разрешение обмена данными по РМ (формируется подключенными к ней модулями в ответ на свой адрес);

CLK — линия тактовой частоты;

INH1 — запрет ОЗУ;

INH2 — запрет ПЗУ.

К линиям RAEN, RSEL, INH1, INH2 должны подключаться передатчики с открытым коллектором, а к линии INT — с ТТЛ-выходом. Назначение и тип передатчиков остальных линий РМ совпадают с системной магистралью.

Блок дешифрации адреса (рис. 5) выполнен на основе ПЗУ D44 и D45. Элементы D38...D41 выполняют предварительную дешифрацию адреса, так как прерывания используются только на РМ, то микросхема D42 во время цикла обслуживания запроса пре-

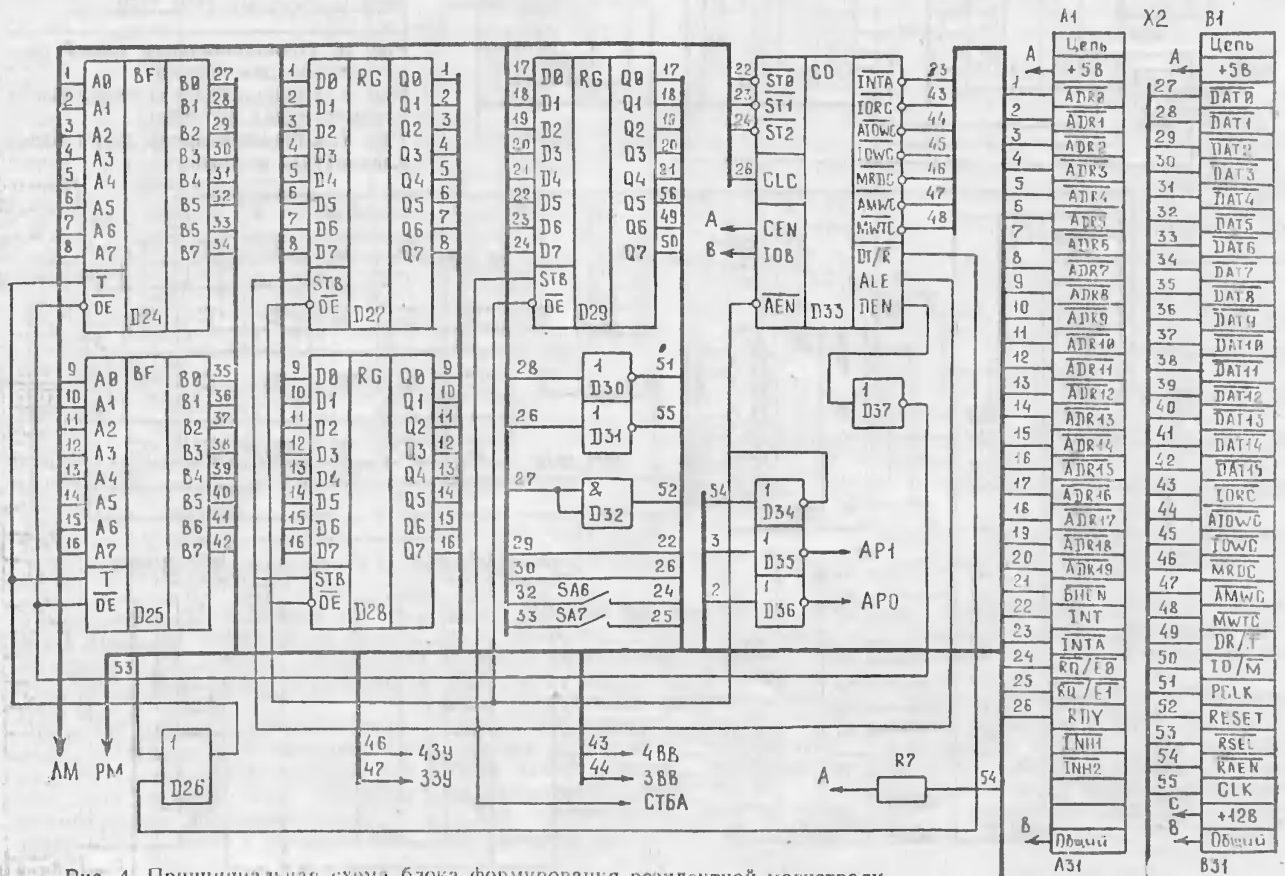
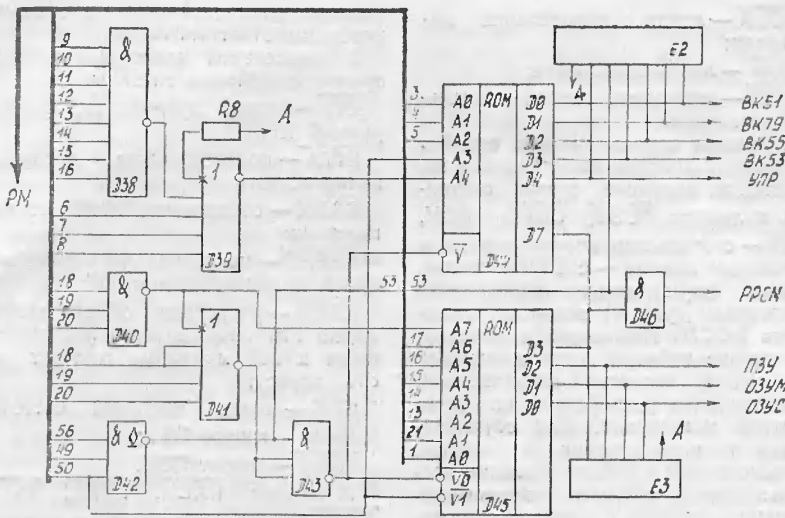


Рис. 4. Принципиальная схема блока формирования резидентной магистрали



ривания формирует сигнал RSEL и запрещает работу устройств микроЭВМ. Элемент D46 выдает сигнал, разрешающий работу системной магистрали при высоком уровне сигнала RSEL и сигналах с выходом D7 и D3 ПЗУ D44 и D45.

Блок запоминающих устройств (рис. 6) построен с использованием ОЗУ (D48, D49) каждое объемом 2К×8 и ПЗУ (D50) — 4К×16 бит. Так как ОЗУ имеет раздельные входы для адреса и данных, его адресные линии подключены к РМ.

Блок клавиатуры и дисплея (рис. 7) выполнен на основе БИС КР580ВВ79. Сканируемая клавиатура построена в виде матрицы 4×8 и содержит 28 клавиш. Клавиша SB3 используется для формирования сигнала КШЦ и в целях упрощения схемы занимает всю восьмую строку матрицы. Элементы D 55, R9, R10, С4 служат для устранения дребезга.

Блок последовательного интерфейса (рис. 8) реализован на основе БИС КР580ВВ51А и КР580ВВ53. Элементы R40...R42, VT10, VT11, С10 позволяют с помощью БИС таймера формировать звуковой сигнал.

Блок параллельного интерфейса (рис. 9) собран на основе БИС КР580ВВ55А. Обеспечивает ввод и вывод информации по каналам А и В. Для усиления сигналов используются микросхемы D70...D72.

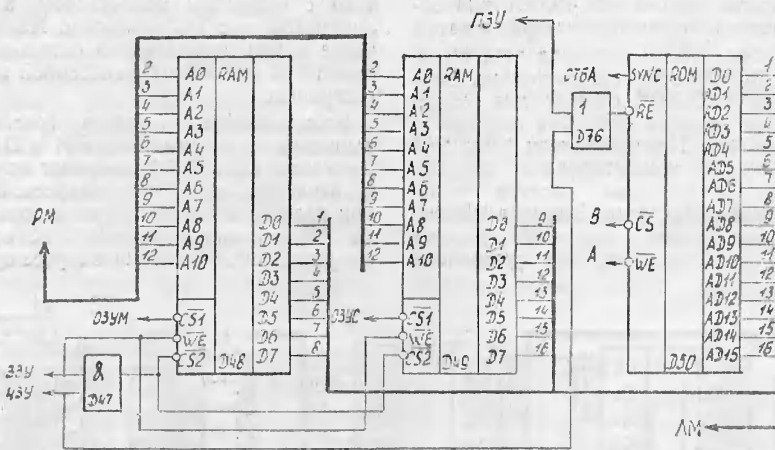


Рис. 5. Принципиальная схема блока дешифрации адреса

Рис. 6. Принципиальная схема блока запоминающих устройств

Рис. 7. Принципиальная схема блока клавиатуры и дисплея

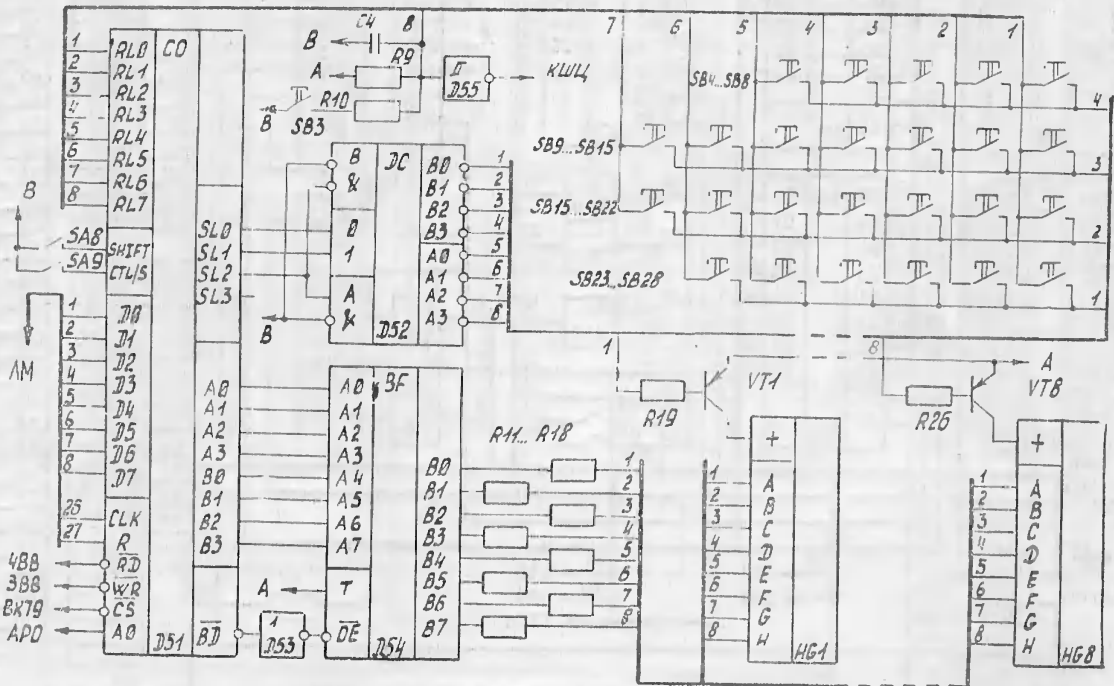


Рис. 8. Принципиальная схема блока последовательного интерфейса

Рис. 9. Принципиальная схема блока параллельного интерфейса

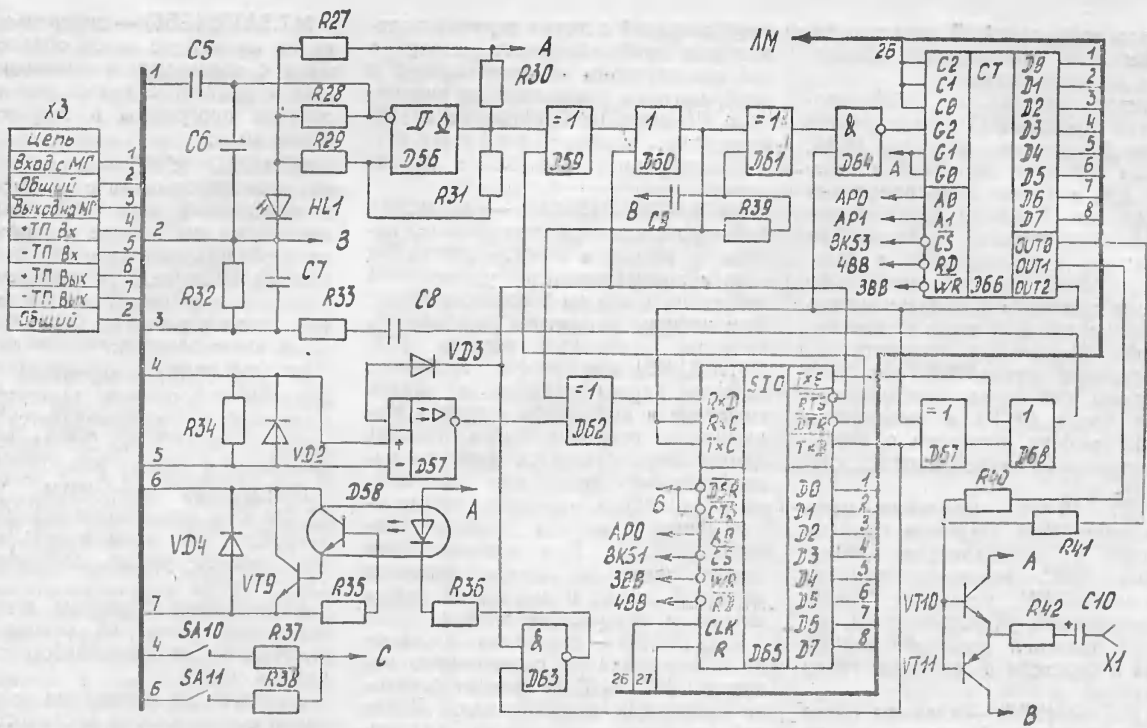


Рис. 8. Принципиальная схема последовательного интерфейса

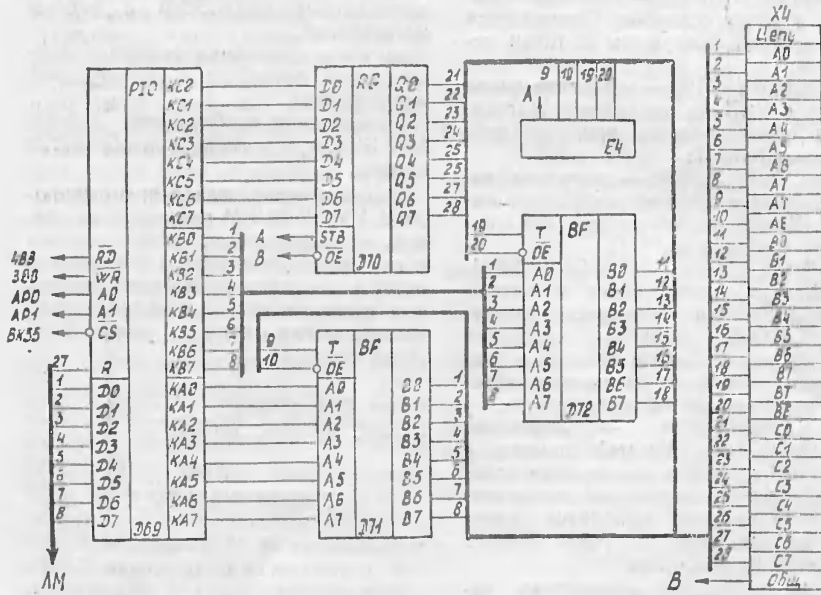


Рис. 9. Принципиальная схема параллельного интерфейса

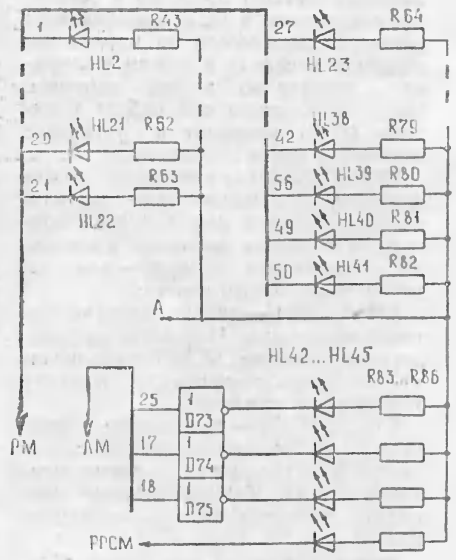


Рис. 10. Принципиальная схема блока адаптации состояния магистралей

Блок индикации состояния магистралей (рис. 10) выполнен на основе светодиодов HL2...HL45. Резисторы R43...R86 служат для ограничения тока.

Режимы работы микроЭВМ задаются резидентным монитором. Он обеспечивает ввод с клавиатуры или магнитной ленты в ОЗУ программ пользователя в машинных кодах, отладку в выполнении в реальном мас-

штабе времени по командам и циклам обращения к магистрали. Монитор позволяет проводить трассировку выполнения программ с возможностью оперативного изменения содержания внутренних программно-доступных регистров МП БИС и памяти, а также выводить информацию на магнитную ленту.

Для обеспечения работы в мониторе введена программа калькулято-

ра, выполняющего арифметические действия с 4-разрядными шестнадцатеричными числами с применением польской инверсной записи.

Монитор располагается в ПЗУ микроЭВМ в адресах FE000H...FEAF7H и использует области ОЗУ с адресами 00000...0000FH и 00E00...00FFFH для временного хранения содержимого регистров пользователя, переменных, области стека и таблицы

векторов прерываний. В качестве системных УВВ используются клавиатура и дисплей модуля.

Монитор представляет собой совокупность программы интерпретации состояний конечного автомата (КА), таблицы описания состояний и переходов КА, а также функциональных подпрограмм. Входным воздействием КА является нажатие клавиши на клавиатуре, в соответствии с которым программа интерпретации изменяет его состояние и производит определенные действия посредством вызова функциональных подпрограмм.

Клавиатура микроЭВМ содержит 16 клавиш для ввода шестнадцатеричных чисел (0...F) и выполнения функций выбора регистров и директив, а также 12 управляющих клавиш:

RESET (SB1) — начальная установка микроЭВМ. Нажатие клавиши приводит к инициализации интерфейсных БИС, расположенных на плате микроЭВМ, установке векторов прерываний, указателя стека, начальных значений регистров пользователя и перехода в состояние ввода директивы;

STOP/DIR (SB2) — клавиша имеет двойную функцию. Нажатие ее во время выполнения программы пользователя (STOP) приводит к останову, сохранению в ОЗУ значений всех регистров процессора на момент останова и переходу в режим модификации памяти по адресу останова. Нажатие клавиши при работе монитора (DIR) приводит к установке состояния ввода директивы;

START (SB16) — выполняет две функции. В режиме ввода директивы используется для запуска программы с адреса останова, а в режиме модификации памяти — для запуска с введенного адреса;

STEP (SB9) — пошаговое выполнение программы. Используется аналогично клавише START, но позволяет выполнить только одну команду при каждом нажатии;

CYCLE (SB3) — выполнение программы по циклам обращения к магистрали. Действует аналогично клавише START. Каждое нажатие приводит к выполнению очередного цикла;

INPUT (SB24) — ввод значений регистров, адреса и данных, просмотр содержимого регистров МП БИС и памяти при увеличении адреса;

ERASE (SB23) — удаление ошибочно набранных цифр, просмотр содержимого регистров и памяти при уменьшении адреса;

↑ — запись набранного числа в арифметический стек. Действует при наборе 4-разрядных шестнадцатеричных чисел и используется при выполнении арифметических операций в польской инверсной записи;

+, -, ×, / — клавиши для выполнения соответствующих арифметиче-

ских операций с двумя верхними элементами арифметического стека с его последующим восстановлением и отображением результата на индикаторе. Деление на 0 приводит к возникновению ошибки;

Монитор имеет следующие директивы:

MEM.MODIF (SB25) — директива байтного просмотра содержимого памяти и записи в память байтов и слов с автоматическим увеличением адреса на 1 или на 2 соответственно. При наборе директивы необходимо выбрать сегментный регистр (CS, DS, ES, SS) или ввести требуемое значение адреса сегмента и задать смещение в выбранном сегменте. Модификация текущего байта (слова) памяти осуществляется набором нового значения (одна или две цифры — для байта, три или четыре — для слова) и вводом (нажатие клавиши INPUT). При возникновении ошибки записи в память выдается звуковой сигнал и выводится действительное содержимое ячейки;

REG (SB26) — директива просмотра и модификации содержимого регистров МП БИС на момент останова программы пользователя. После набора директивы можно указать требуемый регистр;

PGM.CNT (SB27) — директива выхода в точку останова. Применяется для запуска программы с точки останова;

IN.BYTE (SB19) — директива ввода байта с УВВ по указанному адресу. Ввод осуществляется при нажатии клавиши INPUT;

OUT.BYTE (SB20) — директива записи байта в УВВ по указанному адресу. Запись производится при нажатии клавиши INPUT;

IN.WORD, OUT.WORD (SB21, SB22) — директивы ввода и вывода слова из УВВ по указанному адресу;

BLK.MOVE (SB12) — директива перемещения области памяти с заданной длиной и начальным адресом в указанную область памяти;

BLK.CMP (SB13) — директива сравнения двух областей памяти с заданной длиной и начальными адресами. При возникновении несоответствия на дисплей выводится адрес ошибки. Нажатие клавиши INPUT продолжает просмотр;

BLK.FILL (SB14) — директива заполнения области памяти с указанным начальным адресом 8- или 16-разрядной константой;

BLK.SCAN (SB15) — директива поиска заданного байта или слова в области с указанным начальным адресом и длиной. При обнаружении числа отображается его адрес. Нажатие клавиши INPUT продолжает поиск;

MEM.TEST (SB5) — директива разрушающего тестирования области памяти ОЗУ. Работает аналогично директиве BLK CMP;

MT.SAVE (SB6) — директива записи на магнитную ленту области памяти с заданным начальным адресом и длиной, с указанием адреса запуска программы в текущем сегменте кода;

MT.LOAD (SB7) — директива считывания информации с ленты в ОЗУ по выбранному адресу. Если адрес не введен, то считывание осуществляется в область памяти с адресом, указанным на ленте. При ошибке чтения выдается сообщение, а при правильном вводе в регистры CS и IP заносится адрес запуска;

MT.VRF (SB8) — директива верификации информации, записанной на магнитной ленте. Выполняется аналогично MT.LOAD, но без записи информации в память микроЭВМ.

Применение микроЭВМ УМПК-86/ВМ без дополнительных внешних устройств дает возможность в учебном процессе решать следующие задачи:

иллюстрация структуры микропроцессорных систем на основе МПК серии К1810 и взаимодействие ее основных узлов;

исследование концепции использования сопроцессоров и схемотехнических аспектов ее реализации;

изучение особенностей построения одноплатных микроЭВМ и работы магистралей;

изучение специфики сопряжения 8-разрядных устройств с 16-разрядной магистралью;

исследование особенностей работы МП БИС при конвейерной обработке команд.

Использование нескольких микроЭВМ УМПК-86/ВМ совместно со схемой, выполняющей арбитраж доступа к системной магистрали, позволяет изучить архитектуру многопроцессорных систем и способы эффективного распределения задач и ресурсов в таких системах.

Адрес для справок:
103498, Москва, МИЭТ.

ЛИТЕРАТУРА

1. Преснухин Л. Н., Панфилов Д. И., Романенко О. А., Шаронин С. Г. Микропроцессорная лаборатория по изучению микропроцессорных комплектов с фиксированным набором команд // Микропроцессорные средства и системы. — 1985. — № 1. — С. 77—81.
2. Кобылинский А. В., Москалевский А. И., Гемченко В. А. Однокристалльный высокопроизводительный 16-разрядный микропроцессор К1810ВМ86 // Микропроцессорные средства и системы. — 1986. — № 1. — С. 28—34.

Статья поступила 21 апреля 1986 г.

УДК 681.327

О. Б. Тарутин

**ИНТЕРФЕЙС ЧЕТЫРЕХ ВНЕШНИХ УСТРОЙСТВ
СТАНДАРТА ИРПР К МИКРОЭВМ «ЭЛЕКТРОНИКА 60»**

Многие внешние устройства (ВУ) имеют сопряжение стандарта ИРПР. Это позволяет унифицировать плату интерфейса канала микроЭВМ «Электроника 60» к новым ВУ.

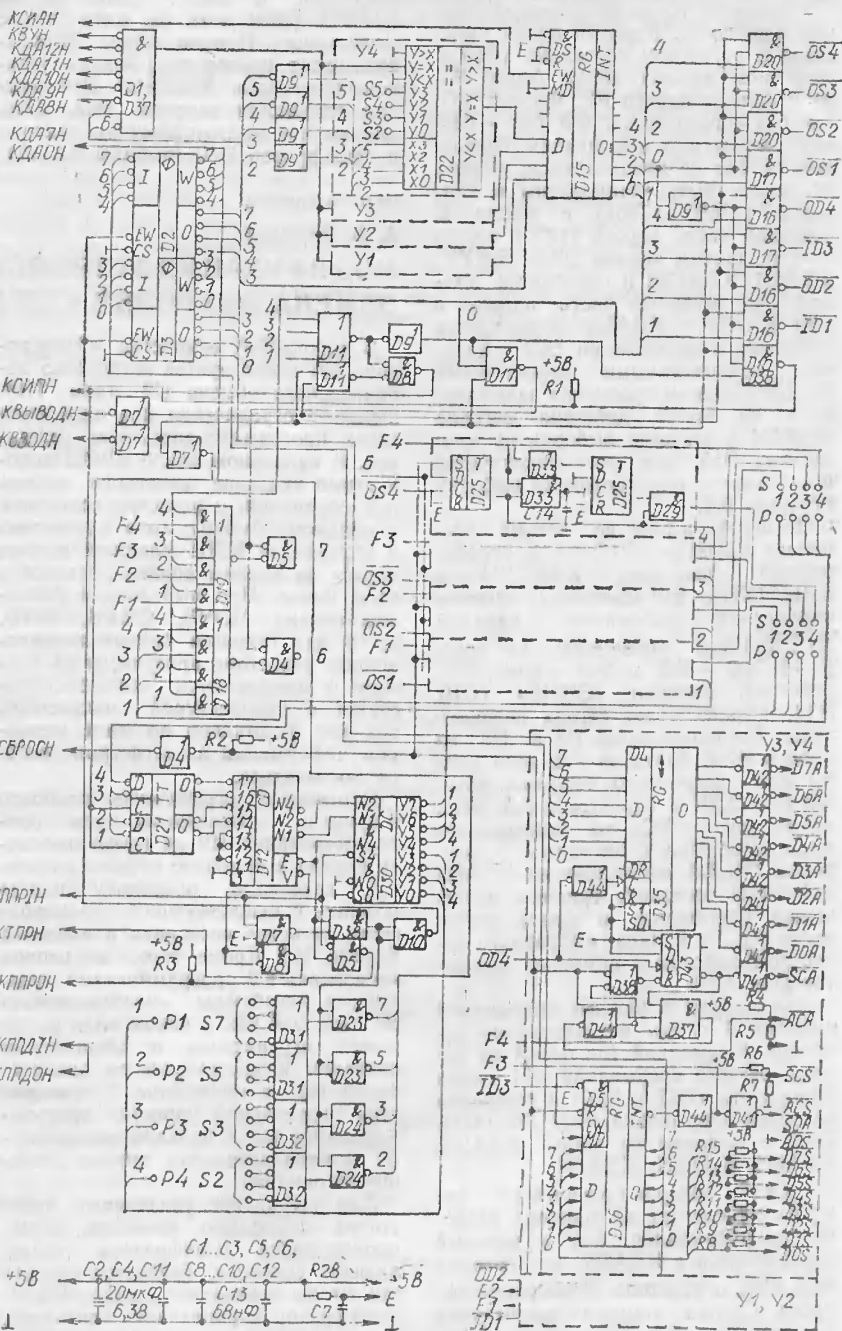
Исключительная простота алгоритма обмена информацией в стандарте ИРПР позволяет разместить интерфейсы к нескольким ВУ компактно на одной плате. Так, если интерфейсы к нестандартным по сопряжению штатным ВУ микроЭВМ «Электроника 60» (считыватель и перфоратор ленты, пультовая пишущая машинка) занимают три полуплаты, последовательный интерфейс дисплея — целую плату, то предлагаемое устройство с четырьмя каналами ИРПР занимает одну полуплату, содержит 46 корпусов микросхем и потребляет ток около 1,3 А от источника +5 В.

К полуплате можно, например, подключить или два дисплея (каждый дисплей имеет один канал ввода и один канал вывода), или дисплей, цифровой вольтметр и АЦПУ и т. д.

Если пользователь микроЭВМ разработывает свое ВУ, то он может сделать его с интерфейсом ИРПР и подключить к свободному гнезду предлагаемой полуплаты. Так, автор изготовил платы программаторов ППЗУ серий К155 и КР556 с двумя разъемами МРН-22-2Р, надеваемыми на разъемы интерфейсной полуплаты.

Приборы, выпускаемые с сопряжением по Британскому стандарту 4421, отличающемуся от ИРПР только напряжением логических уровней (перфоратор DT-105S, прибор для изготовления печатных плат ADMAR и др.), могут быть подключены к предлагаемому устройству через преобразователь уровней.

Схема полуплаты (см. рисунок) содержит 2 канала ввода и 2 канала вывода разрядностью 8 бит без контрольных разрядов, обозначаемых Y1...Y4. Каждый канал ввода использует информационные шины D0S...D7S и управляющие сигналы AOS, ACS, SCS, а каждый канал вывода — информационные шины D0A...D7A и управляющие сигналы SOA, ACA, SCA. Все шины ИРПР — инверсные ТТЛ с открытым коллектором. Максимальный допустимый ток выходных сигналов в активном состоянии (40 мА) обеспечивается вентилями D41, D42. Нагрузка на выходные информационные шины составляет 5 мА, на управляющие — 30 мА за счет резисторов (100 Ом) согласования. Выход на соединительный кабель осуществлен двумя разъемами МРН-22-2В, к каждому из которых подведен 1 канал ввода и 1 канал вывода. Каналы ввода вырабатывают сигналы готовности F1 и F3 от спадающего фронта строба SCS до опроса буферного регистра сигналами ID1 и ID3. В каналах вывода выводимый байт записывается в буфер-



- D1, D11 — К 155 АЕ3; D2, D3 — К 580 АП 16; D4, D5, D6 — К 155 АМ2; D7, D29 — К 133 АМ3; D9, D44 — К 133 АН1;
- D10, D23, D24 — К 155 АА 18; D12 — К 155 ИВ1; D15, D36 — К 580 ИР 12 (3um); D16, D17, D20 — К 133 АА4;
- D18, D19 — К 133 АР3; D21 — К 155 ТМ5; D22 — К 531 СП 1А (4um); D25, D43 — К 133 ТМ2 (5um);
- D30 — К 155 ИД4; D31, D32 — К 133 АА1; D33, D38 — К 155 АМ1 (3um); D35 — К 155 ИР 18 (2um); D37 — К 531 АИЗП;
- D41, D42 — К 155 ПП9 (4um);
- R1, R2 — 430 Ом; R3 — 820 Ом; R4, R6 — 150 Ом (4um); R5, R7 — 390 Ом (4um); R8...R15 — 1,3 к (16um);
- R28 — 470 Ом; C7 — 33 пф; C14 — 680 пф (4um)

Принципиальная схема полуплаты интерфейса четырех внешних устройств стандарта ИРПР к микроЭВМ «Электроника 60»

ный регистр передним фронтом сигналов OD2, OD4, а триггер сброса SCA устанавливается по их заднему фронту. Сигналы готовности каналов вывода F2 и F4 активны, если активен АСА и отсутствует строб SCA.

Управляют всеми четырьмя каналами независимо. Каждый канал представляет регистром состояния (РС) и регистром данных (РД). Адрес РД на 2 больше адреса РС. Формат РС: 7-й бит — готовность, 6-й бит — разряд разрешения прерывания, остальные 14 бит не используются. Адреса РС могут быть установлены в диапазоне 177500—177574 с шагом 4. Старшая часть адреса (1775) поступает по шинам канала ЭВМ КВУН, КДА6Н...КДА12Н и опознается схемой «И». Младшая часть адреса с шиной КДА2Н—КДА5Н сравнивается четырьмя микросхемами D22 с кодами, установленными переключками S2...S5. Сигналы сравнения запоминаются на время действия сигнала КСИАН в регистре выборки на микросхеме D15. Там же фиксируется бит 1 адреса, определяющий выборку РС либо РД.

Дешифратор на восьми элементах «ЗИНЕ» (D16...D20), стробируемый сигналами КВВОДН и КВЫВОДН, вырабатывает сигналы управления выбранным каналом D1...D54 (их мнемоника: I — ввод, S — РС, D — РД, цифра — номер устройства). Элемент «ЗИНЕ» (D10, D38) вырабатывает сигнал переключения формирователей D2 и D3 на ввод в ЭВМ байта из РД, если хотя бы одно устройство выбрано, активен КВВОДН и адресуется РД. Если же адресуется РС, то открываются мощные вентиля с открытым коллектором D4, D5, выдающие на КДА6Н и КДА7Н состояние триггера разрешения прерывания и флага готовности F1...F4 канала, выбранного регистром выборки с помощью элементов D18, D19.

Для работы в режиме прерываний имеются 4 схемы формирования запросов прерываний (по одной на канал), регистр прерываний D21, схема приоритета (D12 и D30) и генератор векторов прерываний (D23, D24, D31, D32) с выходом на шины КДА2Н, КДА3Н, КДА5Н, КДА7Н.

Схема формирования запросов состоит из триггера разрешения прерывания, код в который записывается с КДА6Н по сигналу OS1...OS4, и триггера регистрации запроса. Сигнал запроса прерывания (конъюнкция состояния обоих триггеров) через коммутационную панель поступает в регистр прерываний.

Приоритетный шифратор D12 выдает инверсный код номера запроса с высшим приоритетом на выходы N1, N2, который поступает в двухканальный дешифратор D30.

Код с нестробируемых выводов де-

шифратора Y0...Y3 поступает на подготовку вектора прерывания, который можно установить переключками в виде Y0Y0Y0Y00 (Y означает, что бит может быть установлен). Одновременно приоритетный шифратор D12 выставляет в канал ЭВМ сигнал КТПРН (если хотя бы один запрос существует). Приняв запрос, ЭВМ запрашивает вектор прерывания установкой сигнала КППРН. Этот сигнал стробирует вентиля D23, D24, для выдачи подготовленного вектора и дешифратор D30 (выводы Y4...Y7),

УКД 681.322.074

А. В. Бирюков

МОДИФИКАЦИЯ УСТРОЙСТВА ПАРАЛЛЕЛЬНОГО ОБМЕНА МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

В микроЭВМ семейства «Электроника 60» используется устройство параллельного обмена И2 (или И1), имеющее ограничения при адресации своих программно-доступных регистров. В канальном цикле ВВОД недопустимо указание нечетного адреса при обращении к регистру состояния и входному буферу, хотя формально в интерфейсе МПИ младший разряд адреса не должен анализироваться в этом цикле. При выполнении байтовых команд MOVВ, SMРВ, ВITВ, TSTВ над старшим байтом перечисленных регистров программы по вектору 4 прерываются, что не согласуется с архитектурой микроЭВМ, так как формально по этим командам информация из регистров должна считываться.

Отмеченные ограничения наиболее неприятно проявляются при программировании И2 на языке высокого уровня с помощью средств косвенной адресации, поскольку нужно помнить о существующей неоднородности регистра состояния и входного буфера И2. Кроме того, в период «освоснения» И2 программистами практически неизбежны «необъяснимые» ошибки, так как в явном виде имеющиеся ограничения в техническом описании и инструкции по эксплуатации И2 не приведены. Устранение этих ограничений упростит программирование И2 и избавит пользователя от дополнительных, трудно объяснимых ошибок.

Для устранения указанного недостатка необходимо изменить функционирование дешифратора управляющих сигналов. Требуется сделать так, чтобы в канальном цикле ВВОД дешифратор управляющих сигналов, реализованный на ПЗУ K155PE3, не анализировал младший разряд адреса. Программатор этого ПЗУ является простейшим прибором и в большинстве организаций он, конечно, имеется, поэтому не составит труда запрограммировать в ПЗУ новую информацию и заменить старую микросхему ПЗУ в устройстве И2. Новое

сигнал с которого, пройдя через коммутационную панель, сбрасывает триггер регистрации принятого запроса. Вентиль D5 разрешает прохождение сигнала КППРН далее на КППРН только в случае полного отсутствия запросов в подплате. Коммутационные панели позволяют установить любой взаимный приоритет четырех устройств.

Адрес для справок: 142092, г. Троицк Московской обл., Институт физики высоких давлений им. Л. Ф. Верещагина АН СССР.

Таблица

| Адрес | Данные | | | | | | | |
|-------|--------|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 11 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 12 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 13 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 16 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 17 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 18 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 19 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 20 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 21 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 22 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 23 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 24 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 25 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 26 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 27 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 28 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 29 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 31 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

содержимое ПЗУ приводится в таблице. Отличие его от старого заключается в том, что адресу слова 17 соответствуют номера разрядов 067, а адресу слова 21—567.

После замены ПЗУ не требуется изменение программного и тестового обеспечений. Все программы, которые не используют некорректные ограничения в адресации регистров, идут без изменений с доработанным устройством И2.

Телефон для справок: 972-30-73 (г. Москва)

Статья поступила 30 января 1986 г.

От редакции. Бейсик — алгоритмический диалоговый язык, ориентированный на широкий круг программистов-непрофессионалов. Разработан этот язык в 1965 г., на отечественных ЭВМ культивируется с начала 1970 г. С появлением вычислительной техники третьего поколения (СМ ЭВМ), современных микроЭВМ и микропроцессоров Бейсик превратился в один из наиболее популярных языков программирования. Он широко используется в системах научных исследований, инженерно-технических расчетах, медицинской диагностике и т. д. В ряде стран изучение Бейсика составляет основу школьных курсов программирования для старших классов.

Публикуемые в настоящем номере две довольно разноплановые статьи так или иначе посвящены вопросам стандартизации Бейсика в нашей стране. Редакция обращается к читателям и заинтересованным организациям с просьбой принять широкое участие в обсуждении предлагаемого материала. В первую очередь хотелось бы услышать Ваше мнение о взаимоотношении стандарта с наиболее распространенными версиями, эксплуатируемыми на бытовых и школьных ПЭВМ, полноте и логической завершенности предлагаемых версий, о выборе концепции «ядро + модули» и возможности создания национальной или двуязычной версии Бейсика.

УДК 681.322.068

А. Л. Александров, Е. С. Башмакова, М. Л. Гуткин, А. Б. Либеров О СТАНДАРТЕ ЯЗЫКА БЕЙСИК

В связи с запланированным на XII пятилетку массовым выпуском персональных ЭВМ, для которых Бейсик будет одним из основных языков программирования, вопрос создания государственного стандарта на этот язык становится одним из наиболее актуальных.

Задача стандартизации языка Бейсик ставится не впервые, однако пожалуй единственным существующим стандартом можно считать документ ISO [1], определяющий так называемый «минимальный Бейсик». Вскоре после выхода этого документа (1978) стало ясно, что зафиксированный в нем уровень языка требует при создании практических программ учета архитектурных особенностей конкретной ЭВМ.

Минимальный Бейсик стал ядром многочисленных подмножеств языка, разработанных различными фирмами. Можно выделить два основных направления развития многочисленных диалектов этого языка: *Первое* — Бейсик-системы, встроенные в микроЭВМ, использующиеся в лабораторных, технологических и технических исследованиях. Эти системы в дальнейшем мы не будем рассматривать, так как при их использовании необходимо учитывать особенности аппаратных средств ЭВМ. *Второе* — универсальные Бейсик-системы диалогового типа, обеспечивающие возможность разработки разнообразного программного обеспечения ПЭВМ и предоставляющие пользователю доступ к аппаратным ресурсам. Такие системы наряду с драйверами устройств ввода-вывода являются составной частью ПЗУ, входящего в состав ПЭВМ. Промышленными стандартами для систем этого типа можно считать версии, разработанные фирмой Microsoft (США), а именно — BASIC-80 (для ОС типа CP/M-80), BASICA (для ПЭВМ IBM PC) и MSX BASIC (для ПЭВМ класса MSX).

Язык Бейсик находится в непрерывном развитии и при разработке стандарта надо учитывать этот процесс. Развитие можно проследить, проанализировав возможности шести наиболее распространенных Бейсик-систем, разработанных за рубежом в период 1979—1985 гг. (табл. 1).

BASIC-80 и MBASIC отличаются друг от друга в основном системными вызовами, что объясняется их общим происхождением (фирма Microsoft), и в дальнейшем они будут рассмотрены как одна система. Существуют много модификаций MBASIC для CP/M, которые отличаются основным носителем (НКМЛ, НГМД), объемом занимаемой памяти и числом инструкций, на-

Таблица 1

| Бейсик-система | Фирма-поставщик | Операционная система | Объем занимаемой памяти, Кбайт | Число инструкций |
|----------------|-------------------|----------------------|--------------------------------|------------------|
| BASIC-80 | Intel | ISIS-II | 22 | 112 |
| MBASIC | Microsoft | CP/M-80 | 24 | 122 |
| XYBASIC | Mark Williams | ISIS-II | 16 | 111 |
| BASIC/5 | Hampshire College | CP/M-80 | 7 | 51 |
| BASICA | Microsoft (IBM) | ПЗУ, MSDOS | 40 | 167 |
| MSX BASIC | Microsoft (MSX) | ПЗУ | 32 | 164 |

Таблица 2

| Тип переменных | BASIC-80 | XYBASIC | BASIC/5 | BASICA | MSX BASIC |
|---------------------------------------|----------|---------|---------|--------|-----------|
| Строчковые | + | + | | + | + |
| Числовые | + | + | | + | + |
| Десятичные: цели | + | + | + | + | + |
| с фиксированной точкой | | | | | |
| с плавающей точкой одинарной точности | + | + | + | + | + |
| двойной точности | + | + | | + | + |
| Шестнадцатеричные | + | + | | + | + |
| Восьмеричные | | | | + | + |
| Двоичные | | + | | | + |

Таблица 3

| Язык | Цели | Одинарной точности | Двойной точности |
|-----------|--------------|-----------------------|-------------------------|
| BASIC-80 | -32768+32767 | +1.2E-38+ +3.4E+38 | +2.2D-308+ +1.8D+308 |
| XYBASIC | -32768+32767 | +1.7E-38+ +1.7E+38 | |
| BASIC/5 | -99999+99999 | +1E-128+1E+125 | |
| BASICA | -32768+32767 | +2.9E-39+ +1.7E+38 | +2.2D-308+ +1.8D+308 |
| MSX BASIC | -32768+32767 | +10E-64+10E+64 | +10D-64+ +10D+64 |

пример OBASIC, MBAS и другие. Эти интерпретаторы в данной статье рассматриваются под именем BASIC-80. Наиболее приближенным к возможностям минимального Бейсика является интерпретатор BASIC/5, хотя и здесь набор инструкций несколько шире этого ядра.

Проведем сравнение возможностей, предоставляемых Бейсик-системами в общем, вычислительном, системном, техническом и графическом аспектах. В аспекте общих возможностей будут затронуты следующие особенности систем в языке BASIC: нумерация строк, типы переменных, диапазон данных, имена переменных, комментарии к программам.

Основной особенностью Бейсик-систем является нумерация программных строк. Каждая программная строка должна начинаться с номера, включающего 1...5 цифр. Максимальным номером строки в рассматриваемых интерпретаторах являются следующие числа: в BASIC-80, MSX BASIC и BASICA — 65529, а в XYBASIC и BASIC/5 — 65535.

Во многих Бейсик-системах существуют два типа переменных: числовые и строковые (табл. 2). В BASIC-80, MSX BASIC и BASICA можно конвертировать, т. е. преобразовывать символы в число и наоборот, переменные (целые с плавающей точкой, одинарной или двойной точности) в число любого типа, используя функции CINT, CSNG и CBEL для перевода в другой формат. В табл. 3 указан диапазон десятичных значений, которые могут принимать переменные. Если значения целочисленных переменных выходят за пределы этого диапазона, то они начинают обрабатываться как числа с плавающей точкой. В различных Бейсик-системах по-разному записываются имена переменных, но в каждой из них первым символом имени переменной должна быть буква. Специальный символ, определяющий тип переменной, должен быть последним во всех Бейсик-системах, где есть различные типы переменных. Вторым символом в имени переменной может быть цифра или буква.

В BASIC-80 и BASICA длина имени переменной не должна превышать 40 символов. Оно может содержать буквы, цифры и точку. В XYBASIC имя может быть любой длины, но запоминаются только первые восемь символов. Имя может содержать буквы и цифры. В BASIC/5 — только два символа, причем вторым символом обязательно должна быть цифра. Во всех Бейсик-системах именем переменной не может быть резервированное слово, и только в BASICA оно может быть частью имени.

В BASIC-80, XYBASIC, MSX BASIC и BASICA операторы DEFINT, DEFNG, DEFDBL и DEFSTR определяют переменные, имена которых начинаются с заданных в операторе букв. Например, после выполнения оператора DEFDBL L-P переменные, имена которых начинаются с букв L, M, N, O и P, будут с плавающей точкой двойной точности. Почти во всех Бейсик-системах, кроме BASIC/5, в строке может быть несколько операторов, разделенных двоеточием. Это удобно для написания и отладки программы.

Интерпретаторы языка BASIC могут выполнять арифметические, логические и строковые операции; встроенные функции, условные операторы и операторы цикла; операции с битами; работу с подпрограммами. Все Бейсик-системы могут выполнять арифметические операции и операции отношения. Во всех системах состав операций отношения одинаков: > < = <> >= <=. Минимальный Бейсик задает следующий состав арифметических операций: (—) — унарный минус — (X) — умножение, (/) — деление, (+) — сложение, (—) — вычитание. В BASIC-80, MSX BASIC и BASICA кроме этих операций существуют: (^) — возведение в степень, (\) — целочисленное деление и (MOD) — операция получения остатка от деления. Кроме этого в XYBASIC добавлена (JOIN) — операция конкатенации двух 8-битовых значений в 16-битовое.

Почти во всех Бейсик-системах, за исключением BASIC/5, можно выполнять логические операции. В XYBASIC можно выполнять четыре операции: NOT, AND, OR и XOR. А в BASIC-80, MSX BASIC и BASICA шесть логических операций: NOT, AND, OR, XOR, EQU и IMP. Во всех Бейсик-системах есть возможность работать с массивами. Для описания массивов используется оператор DIM. Один BASIC/5 работает только с одномерными массивами. Для того, чтобы исключить массивы из программы, в BASIC-80 и BASICA существует оператор ERASE. Использование этого оператора позволяет динамически перераспределять память в процессе выполнения программы.

В каждой Бейсик-системе есть встроенные арифметические функции, позволяющие очень легко запрограммировать такие операции как вычисление квадратного корня, логарифма числа, синуса угла и т. п. Наименьшее число функций предоставляет BASIC/5: ABS, COS, INT, RND, SGN, SIN, SQR, TAN.

В XYBASIC эти функции дополнены операторами ATN, EXP, LOG. В BASIC-80, MSX BASIC и BASICA к функциям XYBASIC добавляются: CINT — округление числа, FIX — усечение до целого числа.

При решении некоторых задач требуется, чтобы машина выдавала числа случайным образом. Во всех Бейсик-системах для этой цели предусмотрена функция RND, которая генерирует число, лежащее между 0 и 1 (в XYBASIC между 0 и 32767). В зависимости от аргумента каждый раз при выполнении программы будет генерироваться одна и та же последовательность случайных чисел. Для того, чтобы изменить эту последовательность, надо изменить аргумент функции RND или использовать оператор RANDOMIZE. Если этот оператор поместить в начало программы, то при каждом выполнении будет генерироваться другая последовательность случайных чисел. Во всех

рассматриваемых Бейсик-системах, кроме BASIC/5 и MSX BASIC есть такой оператор.

В рассматриваемых Бейсик-системах, за исключением BASIC/5, пользователю предоставляется возможность самому задать функцию оператором DEF FN. Тип функции определяет какое значение числовое или строковое она возвращает. Эта функция должна быть определена перед первым вызовом.

В дополнение к числовым переменным все рассматриваемые Бейсик-системы, кроме BASIC/5, позволяют использовать строковые переменные, содержащие 0...255 символов. Иногда требуется конвертировать тип данных. Для этого в Бейсик-системах служат следующие операторы:

HEX \square и OCT \square (в XYBASIC, MSX BASIC и функция BIN \square) возвращают строку шестнадцатеричных и восьмеричных (и двоичных) цифр;

STR \square возвращает строку десятичных цифр;

VAL возвращает числовое значение строки, если символы строки — цифры;

CHR \square возвращает код символа, соответствующего заданному числу.

BASIC-80, MSX BASIC и BASICA включают функции CVT, CVS, и CVD, конвертирующие строку в целое число, число с плавающей точкой одинарной или двойной точности и функции MKI \square , MKS \square и MKD \square , конвертирующие числа соответствующей точности в строку.

В рассматриваемых Бейсик-системах для работы с частью строки используются следующие функции: LEFT \square — возвращает левые символы строки, RIGHT \square — возвращает правые символы строки, MID \square — возвращает заданную часть строки.

Функция ASC используется для получения кода ASCII первого символа заданной строки. Функция INSTR возвращает позицию нахождения одной строки в другой.

В XY BASIC для этой цели добавлена функция GET \square , которая позволяет проверить строку на наличие определенного символа. Если этот символ задан, то функция GET \square возвращает его, а если нет, то возвращается пустая строка.

В BASIC-80, MSX BASIC и BASICA при работе с файлами иногда требуется переслать строковые данные из памяти в буфер произвольного файла. Для этого используются операторы LSET и RSET.

Рассматриваемые Бейсик-системы могут работать со строковыми массивами также, как с числовыми. Оператор DIM будет описывать и строковые массивы. Условные операторы и операторы цикла являются как бы стандартной частью и практически одинаковы во всех версиях. Условные операторы имеют один и тот же формат: IF...THEN...[ELSE] (за исключением BASIC/5 и XYBASIC, где используется формат IF...THEN). В BASIC-80 и MSX BASIC применяется также формат IF...GO TO. С помощью оператора IF, организации счетчиков и операторов GO TO можно строить циклы. Для более удобного построения циклов в Бейсик-системах существует оператор FOR...NEXT. В BASIC-80 и BASICA можно организовать цикл и с помощью оператора WHILE...WEND.

Большая часть интерпретаторов не имеет операций над битами, за исключением XYBASIC, а именно они позволяют легко написать программы управления на языке BASIC, а не на языке ассемблера. Указанные ниже функции характерны только для XYBASIC:

TEST — проверка определенного бита целого числа с возвратом значения бита (0 или 1);

SET — установка бита в заданное значение;

RESET — изменение значения бита;

ROTATE — циклический сдвиг целого числа вправо на заданное число позиций. Функции LSHIFT и RSHIFT осуществляют сдвиг целого числа влево или вправо на заданное число позиций;

BIN — представление двоично-десятичного числа в двоичном коде;

BCD — преобразование двоичного числа в двоично-десятичное.

Работа с подпрограммами в рассматриваемых системах может организовываться следующим образом:

— для вызова подпрограммы, которая является частью выполняемой программы, используется оператор GOSUB. Во всех рассматриваемых Бейсик-системах, кроме BASIC/5, с помощью оператора ON...GOSUB можно вызвать одну из подпрограмм в зависимости от значения выражения оператора ON;

— для вызова подпрограммы на машинном языке используется оператор CALL. В BASIC/5 для передачи аргументов в подпрограмму, вызванную оператором CALL, используется функция ARG. В XYBASIC выполняется оператор SCALL, подобный оператору CALL, за исключением того, что значения параметров передаются в и из подпрограммы на машинном языке, непосредственно через регистры. Передаваемые параметры должны быть целочисленными.

— для связи заданной подпрограммы с программой, находящейся в памяти, и передачи ей аргументов в BASIC-80 и BASICA используются операторы MERGE, COMMON и CHAIN;

— для загрузки и сохранения подпрограмм на машинном языке в MSX BASIC, BASICA используются операторы BLOAD и BSAVE;

— в BASIC-80, MSX BASIC и BASICA есть возможность определить оператором DEFUSR начальный адрес подпрограммы на машинном языке, которая позднее вызывается функциейUSR с заданным аргументом.

Бейсик-системы, как правило, не включают в язык средств системного вызова конкретной ОС, проводя операцию с УВВ и с человеком на уровне языка. Одна система от другой отличается инструкциями, реализующими следующие системные возможности: возврат в ОС, загрузка и сохранение созданных программ и файлов, назначение логических устройств, работа с файлами, способы отладки разработанных программ, коммуникационные возможности.

Многие Бейсик-системы отличаются способом возврата из BASIC в операционную систему. Для этого в BASIC-80 (фирмы Intel) существует команда EXIT, в MSX BASIC, BASICA и BASIC/5 используется команда SYSTEM, а в XYBASIC команда CTRL-B.

Сохранение программ во всех Бейсик-системах осуществляется командой SAVE. В BASIC-80, MSX BASIC, BASICA и XYBASIC программы можно сохранить в кодах ASCII. Для этого в команде SAVE указывается параметр A. Если в BASIC-80 и BASICA в команде SAVE указывается параметр P, то программа будет сохраняться в закодированном двоичном формате. Это параметр защиты. Такую программу нельзя распечатать и отредактировать. Способы снять защиту с нет. В XYBASIC программу можно сохранить и в шестнадцатеричном коде, если в команде SAVE указать параметр H.

Во многих системах назначение логических устройств производится: в рамках ОС установлены постоянные имена устройств ввода-вывода. Однако в XYBASIC существует возможность присвоения имен физическим устройствам логическим с помощью оператора ASSIGN. В XYBASIC функция IOBYTE возвращает текущее значение байта ввода-вывода.

Как правило, системы, ориентированные на работу в среде микроЭВМ, работают с дисковыми файлами. В BASIC-80, MSX BASIC и BASICA существует два типа дисковых файлов, последовательные файлы и файлы произвольного доступа (ФПД). В XYBASIC — только последовательные файлы.

Легче создать последовательные файлы, чем ФПД, но они ограничивают гибкость и скорость доступа к данным. С последовательными файлами используются следующие операторы и функции:

- оператор OPEN открывает файл для ввода или вывода;
- оператор CLOSE закрывает файл;
- операторы PRINT # и PRINT USING, а в BASIC-80 и BASICA оператор WRITE # записывают данные в файл;
- операторы INPUT # и LINE INPUT # (LINPUT # в XYBASIC) считывают данные в программу из последовательного файла;

- функция EOF используется для проверки конца файла;
- функция LOC (нет в XYBASIC) возвращает номер сектора;
- команда MARGIN (только в XYBASIC) позволяет форматировать выходной файл, определяя максимальную ширину каждой выходной строки.

С ФПД кроме операторов OPEN и CLOSE, используются следующие операторы и функции:

- FIELD для резервирования пространства в буфере для переменных, которые будут записаны в произвольный файл;
- LSET, RSET пересылают данные в буфер ФПД. Числовые значения должны быть переведены в строковые;

- PUT записывает данные из буфера в файл;
- GET считывает запись из файла;
- LOC возвращает номер текущей записи.

В Бейсик-системах существует возможность стереть файлы, не выходя из интерпретатора. В BASIC-80, MSX BASIC и BASICA эту функцию выполняет команда KILL, в XYBASIC — команда SCRATCH, а в BASIC/5 команды ERA и UNSAVE.

В рассматриваемых Бейсик-системах, кроме BASIC/5, есть команды, которые позволяют сделать следующее:

- AUTO автоматически генерирует номера строк;
- RENUM перенумеровывает номера строк;
- DELETE стирает заданные программные строки;
- CONT продолжает выполнение программы после нажатия CTRL-C или встречи оператора STOP;
- FDIT вызывает строку текущей программы для редактирования, т. е. загружает встроенный редактор.

Когда BASIC встречает ошибку в программе, ее выполнение приостанавливается, и на экране появляется сообщение об ошибке. В BASIC-80, MSX BASIC, BASICA и BASIC/5 сообщение об ошибке имеет следующий вид:

Syntax error in pppp, где pppp — номер программной строки, т. е. указывается тип ошибки и в какой строке встретилась ошибка.

XYBASIC, когда находит ошибку, печатает сообщение с указанием приблизительного расположения ошибки.

В BASIC-80 и BASICA, если обнаруживается синтаксическая ошибка, то режим редактирования включается автоматически и некорректная строка появляется на экране.

Каждая Бейсик-система, кроме BASIC/5, имеет встроенный редактор, который вызывается командой EDIT с указанием номера строки, подлежащей редактированию. BASIC-80, XYBASIC имеют командные редакторы. С помощью команд редактора в исправленной строке, начиная с текущей позиции курсора, можно стереть и вставить N символов, стереть все символы справа или слева от курсора, найти определенный символ, изменить N символов, восстановить первоначальную строку с

выходом и без-выхода из редактора, напечатать остаток строки и т. д.

Другой важной сервисной функцией является обеспечение отладки программ. С помощью трассировки выполнение программы может быть проверено строка за строкой, что позволяет определить точно нахождение ошибки. В BASIC-80, MSX BASIC и BASICA трассировка включается (выключается) командами TRON/TROFF. После команд TRON и RUN на экране дисплея появляются номера строк каждого исполняемого оператора. В XYBASIC трассировка включается (выключается) командами TRACE/UNTRACE. На экране дисплея, кроме номеров строк, появляются сами программные строки и имена переменных.

В XYBASIC команда BREAK позволяет установить точку приостановки по номеру строки или переменной. Если установить точку по номеру строки, то XYBASIC печатает номер строки всякий раз, когда она выполняется, а затем или возвращает человеку управление, или продолжает со следующего шага в программе. Если установить приостановку по имени переменной, то XYBASIC печатает имя переменной и ее новое значение всякий раз при ее изменении. Команда UNBREAK уничтожает точку приостановки.

В BASIC-80, MSX BASIC и BASICA можно обнаружить ошибки с помощью моделирования ошибок и подпрограмм ловушки ошибки. Обычно ошибки останавливают выполнение программы, но если встречается смоделированная ошибка, то оператор ON ERROR GOTO передает выполнение программы по определенному номеру строки и запрещает вывод сообщения об ошибке, отличное от определенного пользователем.

Оператор RESUME обеспечивает повторное выполнение программы после обнаружения и обработки ошибки. Можно использовать функции ERR и ERL, которые возвращают код ошибки и номер строки, где встретилась ошибка. В BASIC/5 и XYBASIC ловушка ошибок включается и выключается с помощью команд TRAP и UNTRAP.

В системах MSX BASIC и BASICA имеются команды, обеспечивающие коммуникационные возможности микроЭВМ. Это позволяет программисту осуществить доступ к каналу связи, не выходя из интерпретатора. Для этого используются следующие команды:

- OPEN COM открывает буфер для ввода-вывода коммуникационных файлов;

- ON COM устанавливает номер строки специальной программы, организующей прерывание для режима коммуникации;
- COM ON/OFF/STOP активизирует и деактивизирует работу по приему/передаче данных;

В системах фирмы Microsoft (BASIC-80, MSX BASIC и BASICA) существуют операторы и функции, которых нет в других рассматриваемых Бейсик-системах:

- SWAP обменивает значения двух переменных;
- VARPTR возвращает адрес переменной или блока управления файлом;
- NAME переименовывает файл;
- ATTRIB (только в BASIC-80 фирмы Intel) меняет атрибуты файла.

Важнейшей особенностью Бейсик-систем является наличие операторов управления внешними устройствами. Во всех системах имеются операторы OUT и IN (INP), аналогичные по своему действию операторам OUT и IN языка ассемблера. С помощью оператора POKE можно записать байт по определенному адресу памяти, а функция PEEK возвращает байт, считанный из определенного адреса. В рассматриваемых Бейсик-системах, кроме BASIC/5, можно приостановить с помощью оператора WAIT выполнение программы пока заданный входной порт машины не выработает байт определенного образа. В дополнение к рассмотренным возможностям в XYBASIC есть функция SENSE, которая позволяет найти значение определенного бита входного порта. В BASIC/5 оператор PUSH заносит адрес во внутренний стек машины, а функция POP считывает значения из внутреннего стека.

Язык BASIC имеет большие возможности для вывода данных на экран дисплея и печатающее устройство. Можно управлять форматом печатаемых данных, форматировать выходную строку:

- TAB позволяет печатать в заданной позиции строки. Используется только в операторах PRINT и LPRINT;

- SPC печатает заданное число пробелов в строке (нет в BASIC/5);

- POS возвращает текущую горизонтальную позицию курсора (нет в BASIC/5);

- CSRLIN возвращает текущую вертикальную позицию курсора (только в MSX BASIC и BASICA);

- SPACE \square возвращает строку, состоящую из заданного числа пробелов (нет в BASIC/5 и XYBASIC);

- NULL определяет число пустых символов, напечатанных в конце строки.

В BASIC-80, MSX BASIC и BASICA оператор PRINT USING и LPRINT USING определяет формат печати числовых и текстовых выражений. Оператор PRINT USING определяет правое выравнивание, знак + или — в начале или конце числа, знак * и т. п.

В BASIC-80 и BASICA существует оператор WRITE, который выводит данные на экран. Различие между операторами WRITE и PRINT в том, что оператор WRITE вставляет запятые между элементами, заключенными в кавычки, оставляет пробел перед положительными числами. В BASICA с помощью оператора LPRINT и функции CHR \square можно управлять шири-

ной шрифта, шириной интервала между строками, регулировать длину строки, устанавливать остоны горизонтального джуйлятора, печатать в одном или в двух направлениях и т. д.

В Бейсик-системах есть возможность влиять на программу с клавиатуры во время ее выполнения. С помощью оператора INPUT можно ввести данные в программу пользователя с клавиатуры во время выполнения программы. Оператор LINE INPUT ставит в соответствие строку данных, введенную с клавиатуры.

В дополнение к этому в MSX BASIC и BASICA существуют следующие возможности:

INKEY \square (есть также в BASIC 80) считывает символ с клавиатуры. При использовании INKEY \square считанные символы не высвечиваются на экране, а передаются в программу. Введение этого оператора позволяет управлять программой, например, остановить ее нажатием определенной клавиши;

KEY позволяет перепрограммировать функциональные клавиши, присваивая им любые нужные значения;

KEY ON/OFF распечатывает или стирает значения функциональных клавиш на служебной строке дисплея;

ON KEY позволяет вызвать заданную подпрограмму в зависимости от нажатия определенной функциональной клавиши во время выполнения программы.

Обычно интерпретатор языка BASIC размещается в ОЗУ, и в памяти находится только одна программа, которую он обрабатывает. В отличие от других интерпретаторов XYBASIC позволяет нескольким программам пользователя быть в памяти одновременно (в ОЗУ или в ПЗУ) и переключает управление с одной на другую. Это позволяет, как интерпретатору, так и программе пользователя находиться в ПЗУ. Команда MOVE TO создает новый образ текущей программы, которая затем может быть зашла в ПЗУ. Команды MOVE TO address копирует программу из одной области памяти (ОЗУ или ПЗУ) в любое место. Команда EXEC позволяет XYBASIC обеспечить доступ программам, находящимся вне рабочей области памяти. Адрес, который определен в команде EXEC, может быть или в ОЗУ, или в ПЗУ. Функции FIRST и LAST возвращают адреса первого и последнего байта текущей программы.

В BASIC-80 (фирмы Intel) существует команда PRUN, которая выполняет программу, записанную в ПЗУ. Адрес программы — целое число.

BASICA и MSX BASIC имеют особые операторы и функции, которых нет в других языках BASIC. В BASICA существует два режима: текстовый и графический. С помощью оператора SCREEN можно управлять заданием режима. С помощью оператора LOCATE курсор устанавливается в заданную позицию на заданной строке.

Используя оператор COLOR, можно задать любые из 16 определенных цветов.

В BASICA и MSX BASIC имеется возможность создать звуковое сопровождение программы. Для этого используются операторы BEEP, SOUND и PLAY. С помощью оператора PLAY можно создать музыкальную фразу. Имеется возможность установить длительность нот, выбрать одну из семи октав и т. д. При получении звука с помощью оператора SOUND задается частота звука и его продолжительность. BASICA имеет специальные операторы и функции, разрешающие ввод с помощью светового пера и джойстика.

Специальные операторы управления внешними устройствами введены и в язык XYBASIC. Так, оператор DELAY позволяет задавать временные последовательности до сотых долей секунды. Оператор ENABLE — контролировать поступление события в реальном масштабе времени.

Особо следует остановиться на графических возможностях Бейсик-систем. В MSX BASIC и BASICA в графическом режиме с помощью специальных операторов можно изменять разрешающую способность, задавать число цветов и т. д. В графическом режиме координаты точки задаются в абсолютной и относительной формах: в виде (x, y), где x — горизонтальная, а y — вертикальная позиция точки на экране (абсолютная форма); определяются относительно местоположения предыдущей точки (относительная форма) STEP (x-смещение, y-смещение).

В графическом режиме используются операторы DRAW, LINE и CIRCLE.

MSX BASIC имеет возможность работать с видеодисплейным процессором (VDP). Для этого используются следующие функции и операторы:

VPOKE — внести значение в видеоОЗУ;

VPEEK — считать значение из видеоОЗУ;

VDP — возврат значения регистра;

BASE — возврат базового адреса таблиц видеоОЗУ.

Видеоисплейный процессор MSX поддерживает работу со «спрайтами» (sprite-определенный пользователем символ или образ). В MSX BASIC можно использовать четыре размера спрайтов, которые задаются в операторе SCREEN. Для работы со спрайтами служат операторы SPRITE \square — определение спрайта, PUT SPRITE — вывод спрайта на экран, ON SPRITE GOSUB — ловушка спрайтов при сопадении, SPRITE ON/OFF/STOP — блокирование и разблокирование ловушки спрайтов.

Результаты анализа показывают, что уровни языка в различных системах, в том числе разработанных одной фирмой, сильно отличаются. Это отличие затрагивает

не только большую часть отдельных инструкций языка, но связано с включением в более поздние версии целых классов инструкций (графика, звук и т. п.). Естественным является поэтому стремление к стандартизации языка с обеспечением максимальной переносимости разработанного программного обеспечения.

Работы по стандартизации языка Бейсик за рубежом наиболее активно ведутся в США силами АНСИ и в Международной организации ISO [2, 3]. Однако, наш взгляд, эта работа не обеспечит переносимости большого числа (десятки миллионов экземпляров) созданных и разрабатываемых промышленных программ для распространенных ПЭВМ класса IBM PC, MSX и др. Кроме того, проекты нового стандарта там пока находятся в состоянии обсуждения.

Используя результаты проведенного анализа, можно сформулировать основные требования к стандарту языка Бейсик:

— реализация всех операторов стандартной версии языка на отечественных ЭВМ;

— возможность развития языка;

— учет зарубежных промышленных стандартов.

Основные концепции проекта государственного стандарта языка Бейсик. Формально переносимые программы на языке Бейсик образуют сравнительно ограниченный класс программ, удовлетворяющих определенным и достаточно жестким требованиям. Вместе с тем существует потребность создания мобильного обеспечения на языке Бейсик для достаточно сложных задач. Кроме того, стандартизация языка Бейсик должна обеспечить поддержку наиболее удачно сконструированных версий языка, соответствующих массовым архитектурам и аппаратному обеспечению ПЭВМ.

Исходя из этих требований было принято решение строить стандарт языка Бейсик по схеме «ядро+модуль». Такая структура хорошо зарекомендовала себя при построении стандарта языка Кобол и используется в разработке нового стандарта языка Фортран-8X [4]. В ядре предполагается описать средства, общие для всех реализаций языка. Каждый модуль будет содержать совокупность проблемно- или архитектурно-ориентированных языковых средств. Ядро составят минимальный Бейсик, определенный стандартом Международной организации по стандартизации в 1984 г.

В стандарте предполагается рассмотреть следующие модули:

расширение основных средств языка;

графические средства языка;

работа с накопителями на магнитных дисках;

работа с накопителями на магнитных лентах;

командный модуль.

По мере развития технических и программных средств предусматривается пополнение как состава каждого модуля, так и набора модулей в целом. Совокупность ядра и модулей позволяет обеспечить гибкость выбора фиксированного подмножества стандарта, в максимальной степени удовлетворяющего конкретным потребностям пользователей и возможностей оборудования. Кроме того, ГОСТ на язык Бейсик планируется согласовать с готовящимся сейчас новым стандартом ISO языка Бейсик тоже за счет выбора подходящего подмножества стандарта [5].

По аналогии со стандартом минимального Бейсика в ядре будут описаны синтаксис и семантика языковых конструкций, приведен перечень обнаруживаемых отклонений от стандарта, обязательный для каждой соответствующей стандарта реализации, а также рекомендация по истолкованию и реализации трудных для понимания фрагментов описания ядра. Таким образом, стандарт языка Бейсик планируется сделать закрытым, т. е. включить в него требования к поведению языкового процессора при обработке отклонений от стандарта [6]. Подобное требование позволит обеспечить унификацию диагностики.

В ядре зафиксирована также совокупность свойств

языка, которые не определяются стандартом или зависят от реализации. Ясно, что использование таких средств в программах снижает их переносимость. Поэтому предполагается при описании стандарта включить в него рекомендации по обеспечению мобильности программ для включенных в стандарт языковых конструкций.

В целом основные задачи, связанные с разработкой стандарта языка Бейсик, включают формирование номенклатуры и содержания модулей описания стандарта, выработку адекватной русской терминологии, учет требований Государственной системы стандартизации.

Одним из критериев формирования состава модулей являлось обеспечение преемственности и мобильности программного обеспечения, создаваемого на языке Бейсик. В первую очередь это относится к реализации языка Бейсик для ПЭВМ, так как этот вид ЭВМ является наиболее массовым и наиболее подверженным изменениям (по составу и по характеристикам). Установив заранее синтаксис и семантику операторов и функций, можно ожидать, что программы, разработанные в настоящее время на некоторых ЭВМ, уже имеющих такие возможности, в будущем смогут работать на всех ЭВМ.

При выборе модулей и определении их состава учитывались также зависимость их от состава и характеристик периферийных устройств и характеристик самой ЭВМ. Ниже приводится предлагаемый состав перечисленных выше модулей.

Модуль расширения основных средств.

1. Набор символов включает дополнительно строчные буквы латинского алфавита, а также прописные и строчные буквы русского алфавита. Это расширение относится только к содержимому примечаний и строковых констант.

2. Увеличивается состав числовых констант, они могут быть целыми, вещественными с одинарной точностью и вещественными с двойной точностью. Целые константы могут записываться в десятичной, шестнадцатеричной (с префиксом &H) и двоичной (с префиксом &B) системах счисления. Константы с одинарной точностью могут содержать в своей записи до шести цифр или экспоненту с буквой E, а константы с двойной точностью — до 14 цифр или экспоненту с буквой D. Кроме того, для явного задания типа константы можно использовать спецификаторы типа: ! для констант с одинарной точностью и # для констант с двойной точностью.

3. Имена переменных могут состоять из произвольного числа символов (букв и цифр). Различаются имена по первым двум символам, причем первый символ — обязательно буква. Имя может заканчиваться спецификатором типа переменной: % — целая переменная, ! — переменная с одинарной точностью, # — переменная с двойной точностью, Q — строковая переменная. Спецификатор типа не учитывается при подсчете числа символов в имени.

4. Расширяется состав арифметических операций: добавляется деление нацело (обозначается \) и вычисление остатка от деления нацело (обозначается MOD).

5. Операции отношения можно выполнять не только над числовыми, но и над строковыми переменными.

6. Вводятся логические операции: отрицание (NOT), и (AND), или (OR), исключающее или (XOR).

7. Добавлены или расширены следующие операторы: ON выражение GOSUB номер строки; IF выражение THEN номер строки 1 (или оператор 1) номер строки 2 (или оператор 2); число размерностей массивов в операторе DIM ограничивается только длиной строки;

для улучшения структурированности программ добавлен оператор WHILE выражение список операторов WEND;

введен оператор спецификации типа DEF INI/SNG/DBL/STR;

оператор замены части строки MID Q (s1, n[, m]) — s2;

оператор обмена значений переменных SWAP s1, s2; при определении функции пользователя количество формальных параметров не ограничено;

оператор изменения содержимого памяти POKE n, t; кроме простого вывода, возможен форматированный вывод с помощью оператора PRINT USING;

для вывода на печатающее устройство введены операторы LPRINT и LPRINT USING;

в оператор ввода INPUT можно включать подсказку; для ввода строк целиком добавлен оператор LINE INPUT;

8. Существенно расширен состав встроенных функций:

получение кода первой буквы строки ASC (x Q);

получение кода символа, соответствующего данному числу CHR Q (N);

получение строки, представляющей шестнадцатеричное значение десятичного аргумента HEX Q (N)

ввод символа из буфера клавиатуры INKEY Q;

ввод строки из N символов INPUT Q (N);

поиск первого вхождения одной строки в другую INSTR Q (I, x Q, y Q);

выделение левой части строки LEFT Q (x Q, I);

выделение правой части строки RIGHT Q (x Q, I);

выделение подстроки MID Q (n Q, I, J);

определение длины строки LEN (x Q);

чтение содержимого байта оперативной памяти PEEK (I);

получение строкового представления числа STR Q (x);

получение численного значения строки VAL(x Q).

9. Для отладки программ, написанных на языке Бейсик, предлагаются следующие средства операторов:

управления трассировкой TRON и TROFF;

псевдопеременные ERR (содержит код ошибки) и

ERL (содержит номер строки, где произошла ошибка);

вызова подпрограммы обработки ошибок ON ERROR

GOTO;

возврата из подпрограммы обработки ошибок

RESUME;

10. Операторы управления прерыванием, вызванным нажатием ключа:

разрешение/запрещение прерывания по ключу KEY ON/OFF/STOP;

обращение к подпрограмме обработки прерывания ON KEY (n) GOSUB;

Модуль графических средств. Графические средства существенно зависят от характеристик аппаратуры.

В первую очередь это относится к разрешающей способности дисплея. Для обеспечения полной переносимости программ следовало бы выводить изображения в условных координатах (как это рекомендовано в NAPLPS), но в настоящее время это невозможно реализовать аппаратно, а программная реализация потребует большого объема памяти. Поэтому пока предлагается в качестве стандартного принять дисплей с разрешающей способностью 256×192 точек с вершиной координат в левом верхнем углу. Если программы позже будут выполняться на ЭВМ с более совершенным дисплеем, то изображение будет занимать часть экрана, но переносимость сохранится. Минимальное число цветов для цветного монитора должно быть не менее 16.

При описании графических операторов будем использовать следующие обозначения: X, Y — абсолютные координаты точки (относительно вершины координат); STEP — относительные координаты точки (отсчет от текущего положения графического курсора); K — номер цвета изображения; L — номер цвета фона; M — номер цвета границы (бордюра).

Список графических операторов:

установки режима SCREEN I, где I — номер режима; установки цвета COLOR [K] [, L] [, M]; установки цвета точки PSET [STEP] (X, Y) [, K]; сброса цвета точки PRESET [STEP] (X, Y); изображение прямой линии и прямоугольника; LINE [[STEP] (X1, Y1)—[STEP] (X2, Y2) [, K] [, B ка, а F — параметр заливки прямоугольника указанным цветом;

изображение окружности, эллипса, дуги CIRCLE [STEP] (X, Y), R[, K[, P[, Q[, O]]], где R — радиус, P — начальный угол в радианах, Q — конечный угол в радианах, O — эксцентриситет; закрашивание области сплошным цветом PAINT [STEP] (X, Y) [, K] [, M];

Графические функции:
определение координат графического курсора X=POS и Y=CSRLIN;
определение номера цвета точки K=POINT (X, Y).

Для облегчения создания сложных графических изображений рекомендуется в состав операторов Бейсик включить оператор DRAW, реализующий графический макроязык (ГМЯ). Формат оператора: DRAW s, где s — строковое выражение, содержащее команды ГМЯ. Состав и синтаксис команд ГМЯ в настоящее время уточняется.

В разных версиях языка Бейсик существуют операторы управления перемещаемыми графическими объектами. В частности, в ЭВМ класса MSX для этой цели используются операторы для работы со «спрайтами» — определяемыми пользователями символами или образами, которые могут выводиться на экран и перемещаться по нему. В ЭВМ других классов аналогичные действия можно выполнять с помощью операторов GET и PUT. Авторы считают, что вопрос стандартизации этого класса операторов в настоящее время преждевременен.

Модуль работы с накопителями на гибких магнитных дисках (НГМД). При работе с программными файлами, находящимися на диске используются следующие команды:

LOAD «имя файла» — загрузить файл с диска;
SAVE «имя файла» — сохранить файл на диске;
KILL «имя файла» — стереть файл с диска;
NAME «имя файла 1» AS «имя файла 2» — переименовать файл;

FILES — распечатать каталог диска;
При работе с НГМД возможны два режима работы с файлами данных: с последовательным и произвольным доступом.

Рекомендуется следующий состав операторов:
Открытие файла OPEN F [FOR R] AS [#]N [LEN-L] где F — имя файла, R — режим (OUTPUT, INPUT, APPEND, если режим не задан, то открывается файл с произвольным доступом), N — номер файла (число в диапазоне от 1 до 15), L — длина записи (для файлов с произвольным доступом);

закрытие файла CLOSE #N1, #N2, ...;
удаление файла KILL F;
переименование файла NAME F1 AS F2 где F1 — старое имя, а F2 — новое имя;
операторы и функции для файлов с последовательным доступом:

чтение данных из файла INPUT # N, s или LINE INPUT # N, v где N — номер файла, s — список ввода, а v — строковая переменная;

запись данных в файл PRINT # N, s где s — список вывода;

функция определения конца файла EOF (#N) =—1, если достигнут конец файла;
операторы и функции для файлов с произвольным доступом:

выделение областей для переменных FIELD [#]N, dASv [, ...]; где d — размер поля в байтах, а v — имя поля (строковая переменная);

чтение в буфер из файла GET N, R, где R — номер записи;

запись в буфера в файл PUT N, R, где R — номер записи;

пересылка данных в буфер с выравниванием вправо RSET v, s или влево LSET v, s, где v — строковая переменная, а s — строковое выражение;

функции преобразования строк в числа CVT (2-байтовая строка); CVS (4-байтовая строка), CVD (8-байтовая строка);

функции преобразования чисел в строки MKI \square (целое выражение); MKS \square (выражение с одинарной точностью), MKD \square (выражение с двойной точностью).

Модуль работы с магнитными лентами. Если в комплект ЭВМ входит накопитель на магнитной ленте, то рекомендуются следующие операторы:

MOTOR ON/OFF — включить / выключить мотор;
LOAD «имя файла» — загрузить файл с кассеты;
CSAVE «имя файла» — сохранить файл на кассете.

Командный модуль. Основные команды:
удаление программы NEW;
вывод текста программы LIST [[N1]—[N2]];
запуск программы на выполнение RUN;
включение / выключение трассировки программы TRON/TROFF.

Приведенный выше состав операторов и функций является основой разрабатываемого в настоящее время проекта Государственного стандарта языка Бейсик.

Адрес для переписки: Москва, ул. Вавилова, д. 30/6, Институт проблем информатики АН СССР, Гуткину М. Л.

ЛИТЕРАТУРА

1. ISO (1978) Draft Standard for Minimal Basic ISO/TC 97/SC S # 447 Part 1, DP6373 г.
2. R. Anderson, The Proposed ANSI Basic Standard Byte. Febr. 1983.
3. И. Махачка. Бейсик — ретроспектива и перспектива будущего // Программирование. — 1985. — № 5. — С. 55 — 63.
4. Меткалф М. Оптимизация в Фортране. М.: Мир, 1985.
5. Государственная система стандартизации. М.; Изд-во стандартов, 1977.
6. Александров А. Л., Бабенко Л. П., Кауфман В. Ш., Ющенко Е. Л. Проблемы стандартизации языков программирования // Управляющие системы и машины. — 1983. — № 4. — С. 14—19.

Статья поступила 9 июня 1986 г.

НА КНИЖНОЙ ПОЛКЕ

Корчак А. Е. Языки программирования Бейсик для микроЭВМ. — МЦНТИ, МНИИПУ, 1987.

Рассматривается класс наиболее распространенных языков программирования для микроЭВМ различных типов (языки Бейсик). Проводится сравнительный анализ свойств языков Бейсик с точки зрения функциональных возможностей, особенностей реализации и областей применения. Подробно рассматриваются языки Бейсик, широко применяемые на современных 8- и 16-битовых микроЭВМ (М — Бейсик, Си — Бейсик, Бейсик — А, MSX — Бейсик, Бейсик/F и др.). Даются рекомендации по выбору языка и сферы его применения. Обсуждаются проблемы стандартизации языка Бейсик.

Заказы на издания МЦНТИ принимаются магазином № 93 «Книга-почтой» Москнизи по адресу: 117168, Москва, В-168, ул. Кржижановского, 14, корп. 1.

А. Е. Корчак

БЕЙСИК/Fs — ЯЗЫК ПРОГРАММИРОВАНИЯ ШКОЛЬНЫХ И БЫТОВЫХ КОМПЬЮТЕРОВ

Предлагается проект единого языка программирования Бейсик/Fs для разных типов школьных и бытовых компьютеров, который обеспечит переносимость программ, даст возможность эффективно и единообразно использовать графические, звуковые и игровые возможности персональных компьютеров (ПК), облегчит обучение основам программирования.

В настоящее время наибольшую популярность приобрела версия Бейсика, разработанная фирмой Microsoft для ПК фирмы IBM [1]. Этой же фирмой разработан MSX BASIC [2], реализованный на бытовых компьютерах типа MSX ряда японских фирм. К сожалению, эти варианты Бейсика значительно отличаются друг от друга, хотя и созданы одной фирмой.

Версия Бейсика для IBM PC полностью включает в себя подмножество (минимальный Бейсик), которое соответствует международному стандарту ISO [3]. Эта версия имеет развитые средства для работы с графическим дисплеем, громкоговорителем, световым пером, игровыми пультами (мышь, джойстик, ракетка). С точки зрения использования Бейсика для профессиональных применений в нем есть ряд недостатков: невозможность создавать модули программы с локальными данными, отсутствие ряда управляющих структур, присущих современным языкам программирования.

MSX BASIC построен на базе версии 4 языка Бейсик фирмы Microsoft, разработанной в середине 70-х годов. MSX BASIC имеет ряд отличий от ISO-стандарта: правила преобразования типов, выполнение операторов FOR, ON, функции TAB, отсутствие оператора RANDOMIZE. Нет развитых управляющих структур, не включен оператор WHILE...WEND, появившийся еще в версии 5 (1979 г.). Неудачны с точки зрения непрофессионального пользователя правила задания имен переменных: допускается любая длина имени, но значащими являются первые два символа. Это приводит к неожиданному для пользователя появлению одинаковых переменных. Графические и звуковые средства языка MSX BASIC в значительной степени ориентированы на аппаратные особенности конкретного ПК (графические режимы, спрайты, непосредственный доступ к регистрам видеопроцессора и звукоинтегратора). По этим причинам MSX BASIC не может претендовать на роль единого языка учебных ПК.

Предлагаемый в статье язык Бейсик/Fs является подмножеством языка программирования Бейсик/F [4, 5], разработанного на Рижском производственном объединении ВЭФ им. В. И. Ленина. Буква s в названии языка появилась из таких ключевых слов как subset (подмножество), school (школа), self — taught person (самоучка).

Язык Бейсик/Fs — для изучения программирования в школе, техникуме, вузе или самостоятельно; для реализации учебных, игровых, тестовых, тренажерных программ; для использования на профессиональных ПК, где затруднено применение языка Бейсик/F (один дискет малой емкости, медленный обмен с диском), а возможностей языка Бейсик/Fs достаточно для решения экономических задач, инженерных расчетов, для быстрой разработки программ управления внешними объектами, подключенных к ПК, в лабораторных работах, научных, технологических экспери-

ментах, где необходима частая и быстрая коррекция управляющих программ.

Преимущества использования языка Бейсик/Fs на школьных и бытовых компьютерах:

развитые графические, звуковые и игровые средства, не зависящие от аппаратных особенностей ПК, независимость от разрешающих способностей дисплея, что достигается введением мировых координат;

легкий переход к использованию языка Бейсик/F на профессиональных ПК;

определенная совместимость с языком Бейсик фирмы Microsoft, особенно в графической части. Конвертор исходных текстов облегчит перенос программ на язык Бейсик/Fs;

возможность постановки на любой ПК путем пере- программирования модуля интерпретатора, в котором сосредоточены все аппаратно-зависимые программы системы;

реализация всех управляющих структур, соответствующих идеологии структурного программирования, облегчит переход от изучения алгоритмов на базе E-языка [6] к конкретному программированию на языке Бейсик.

Рассмотрим элементы языка Бейсик/Fs.

Данные. Язык Бейсик/Fs поддерживает три типа данных: вещественные, целые и символьные. Целые константы задаются в десятичном (—10%), 16-ричном (OFEH) или двоичном (101B) виде. Символьная константа записывается в текстовом виде или как последовательность 16-ричных или двоичных кодов, например «83A8B3D2»H. Такая форма записи позволяет удобнее задавать графические образы по сравнению с языком MSX BASIC, где предыдущая запись выглядит так

$$\text{CHR}\langle\langle\&H83\rangle\rangle + \text{CHR}\langle\langle\&H48\rangle\rangle + \text{CHR}\langle\langle\&HB3\rangle\rangle + \\ + \text{CHR}\langle\langle\&HD2\rangle\rangle$$

Имя переменной, массива, нестандартной функции начинается с буквы и состоит из букв, цифр, знаков подчеркивания, числом не более 31. Тип имени задается знаками ! (вещественный), % (целый), $\langle\langle$ (символьный) в конце имени, либо неявно по первой букве имени, если она задана операторами определения типа DCLINT, DCLSTR, DCLFLP. Для целой и символьной переменной в операторе DIM можно задать максимальную длину, отводимую под ее значение. Целая переменная может иметь длину 1 или 2 в зависимости от диапазона изменения значения: —128...127 или —32768...32767. Массивы могут быть одномерными и двумерными. В языке Бейсик/Fs семь арифметических операций (\wedge */DIV MOD + —), четыре логических (NOT AND OR XOR), шесть операций отношения (= < > < > = <=), операция сцепления строк (&).

Базовые операторы. Для задания общего вида оператора используются следующие обозначения: e, x, y, g, c, p — выражения, v — переменная, m — метка (число или идентификатор), [] — элемент внутри скобок может отсутствовать, ... — любое количество элементов, стоящих слева от этого знака, | — выбор одного из элементов, разделяемых этим знаком.

[LET], v, ... = e — присваивание значения переменным

! | REM <текст> — комментарий

GOTO m — переход на указанную строку

GOSUB m — вызов подпрограммы

RETURN — возврат из подпрограммы

DEF FN <имя> — определение нестандартной функции

[(v, ...)] = e
 IF e THEN m | <оператор> | ELSE m | <оператор> | BEGIN
 IF e THEN BEGIN
 ELSE m | <оператор>
 BEGIN

END IF — операторы для организации выбора одной из двух альтернатив

FOR v = e TO e
 [STEP e]

NEXT [v] — организация цикла по управляющей переменной

FOR [WHILE e]
 NEXT [WHILE e] — задание условного цикла

EXIT FOR [e] — выход из одного или нескольких вложенных циклов

SELECT e
 CASE <константа> [TO <константа>] | <операция отношения> <константа>, ...
 CASE ELSE
 END SELECT — организация выбора одной из нескольких альтернатив

ON e GOTO | GOSUB m, ... — стандартное средство Бейсика для выбора одной из нескольких альтернатив

INPUT [;] [(подсказка),] v, ... — ввод значений с клавиатуры.
 Точка с запятой не переводит курсор в начало следующей строки после ввода значений.

READ v, ...
 DATA <константа>, ...
 RESTORE — организация работы с блоком данных

DIM v [(макс. длина) [(целое) [(целое)]]], ... — задание максимальной длины переменной и границ массива

RANDOMIZE — генерация начала новой непредсказуемой последовательности псевдослучайных чисел, которые станут значениями функций

END — конец выполнения программы

STOP — останов выполнения программы

DCLFLP | DCLINT | DCLSTR | <-буква> [-<буква>], ... — неявное задание типа

TRACE [OFF] — задание и отмена режима трассировки

SWAP v, v — обмен значений переменных

CALL e, ... — вызов подпрограммы в машинах кодах

Средства ввода-вывода. В языке Бейсик/Fs реализован последовательный ввод и вывод данных между программой и внешним носителем (диск, лента). Оператор ASSGN назначает файлу данных на внешнем носителе некоторый номер, используемый в дальнейшем для обозначения этого файла:

ASSGN # n, e — назначение файлу номера

ASSGN # n, * — отмена назначения

Имя файла зависит от операционной среды и обрабатывается физическим модулем интерпретатора. Открытие файла выполняет оператор

OPEN # n, TYPE = IN | OUT | NEW

Для некоторых внешних устройств типы OUT (дописывание к концу файла) и NEW (вывод с предварительной очисткой) эквивалентны. Для ввода и вывода предназначены операторы:

READ # n, v, ... — чтение данных из файла

WRITE # n, e, ... — запись данных в файл

Оператор CLOSE # n закрывает файл. Оператор CHAIN e удаляет программу из памяти, загружает новую программу из указанного файла и передает ей управление.

Работа с дисплеем и принтером. В языке Бейсик/Fs вывод на экран дисплея может идти в двух режимах: текстовом и графическом. В текстовом режиме значения выводятся оператором PRINT. В графическом режиме отображается графическая информация, а если позволяет аппарататура, то и текстовая.

В текстовом режиме экран состоит из k строк по m позиций в каждой (нумерация с 1). В каждой позиции может находиться изображение символа в коде KOI-8. Для позиции определены цвет изображения, цвет фона и признак мигания. Можно установить курсор в любую позицию, получить его координаты и атрибуты позиции.

Текстовый режим задается оператором TEXT p. Поддерживается несколько текстовых режимов, задаваемых параметром p, которые отличаются числом строк и позиций в строке. Рекомендуется использовать p=0 при 20 (16) символах в строке, p=1 при 40 (32), p=2 при 80 (64).

Оператор COLOR [c1], [c2], [c3] задает цвета изображения (c1), фона (c2) и рамки экрана (c3).

Оператор ALST C | L определяет устройство вывода: экран дисплея или принтер.

Оператор PRINT содержит выражения и функции печати, отделенные друг от друга разделителями: запятой, точкой с запятой, двоеточием. Строка печати разделена на зоны одинаковой длины. Запятая переводит курсор в начало следующей зоны, точка с запятой печатает пробел, а двоеточие не перемещает курсор.

Функции печати:

TAB (e) — установка курсора в указанную позицию текущей строки

SPC (e) — сдвиг курсора влево или вправо

CRT (x, y) — установка курсора в указанную позицию экрана

BIN (e) — вывод в двоичном виде

HEX (e) — вывод в 16-ричном виде

PAGE — очистка экрана или перевод листа бумаги

TIM (e) — временная задержка

FMT (lw | Fw.d | Ew.d | Aw | S) — задание формата печати, где

УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ УСТАНОВОК КОМПЛЕКСНОЙ ПОДГОТОВКИ ГАЗА

М. А. Корженко, М. Д. Хныкин

В СПКБ «Промавтоматика» (г. Краснодар) разработан микропроцессорный управляющий вычислительный комплекс, на базе которого в 1985 г. внедрены в промышленную эксплуатацию автоматизированные системы управления технологическими процессами установок комплексной подготовки газа (АСУ ТП) на Уренгоском газовом месторождении.

Основой управляющего вычислительного комплекса АСУ ТП УКПГ является вычислительная управляющая система 15 ВУМС 28-025.

Программное обеспечение микропроцессорной системы представляет собой пакет программ, реализующий информационные (сбор и первичную обработку от датчиков аналогового типа; контроль соблюдения технологических границ; расчет технико-экономических показателей); представление оперативной и технико-экономической информации на средствах

отображения), управляющие (анализ заданного технологического режима; формирование управляющих воздействий; выдачу управляющих воздействий на объект), вспомогательные функции (контроль работоспособности оборудования микропроцессорной системы и устройства связи с объектом; локализацию и сигнализацию неисправностей; восстановление конфигурации вычислительной системы).

Все программы пакета, в зависимости от выполняемых функций, делятся на три основных комплекса программ (КП): организующие (ОПО); функциональные (ФПО); специальные (СПО).

КП ОПО является ядром пакета и выполнен в виде многофункциональной мультипрограммной ОС реального времени, резидентной в ППЗУ, имеющей самостоятельное применение («MIKROS-D»). КП ФПО

реализует информационные функции системы, КП СПО реализует собственно функции АСУ ТП (расчет ТЭП, управляющие функции).

Вся информация, используемая программами пакета, хранится централизованно в базе данных, построенной по сегментно-иерархическому принципу и ориентированной на применение в распределенных (многопроцессорных) системах. Все компоненты пакета и статические элементы базы данных содержатся в ППЗУ.

Пакет является параметрической, модульной программной системой, настраиваемой на конфигурацию объекта и вычислительной системы.

Генерация версии пакета для конкретной АСУ ТП производится на инструментальной мини-микроЭВМ под управлением операционной системы Рафос (Фодос) в диалоговом режиме. Пакет выполнен на языке ассемблера и имеет следующие характеристики: объем ОПО ≈ 10 К байт, СПО ≈ 20 К байт, ФПО $\approx 6,5$ К байт; обрабатываемых аналоговых датчиков — до 512; исполнительных механизмов (регуляторов) — до 256; сводок и сообщений — до 30; внешних устройств — до 15; процессоров — до 3.

Телефон для справок: 54-05-18, г. Краснодар



ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР ПРАВЕЦ 16

ПРАВЕЦ 16 — средство для построения локальных сетей, а также систем автоматизации проектирования и управления технологическими процессами.

Основные технические характеристики:

Оперативная память 256К байт с возможностью расширения до 640К байт

Постоянная память 40К байт

Внешняя память 2 двухсторонних мини - флоппидисковых устройства с двойной плотностью записи

Микропроцессор 16бита/4,77 МГц 18088

Формат текста и изображения 25 рядов по 40 или 80 символов

Программное обеспечение совместимый с IBM-PC

Языки программирования Бейсик, Паскаль, Кобол, Фортран

ПРАВЕЦ 16Н — имеет модульную структуру и программно совместим с IBM-PC. В отличие от других представителей этой серии, данный персональный компьютер объединяет в одной общей коробке монохроматический монитор и контроллер с высокой разрешающей способностью 720×348 точек. Имеется возможность подключения сопроцессора 18087, параллельного интерфейса для печатающего устройства. В текстовом режиме дополнительный видеоконтроллер разрешает работу с 25 текстовыми рядами и 80 знаками в ряду, при этом каждый знак размещен в поле из 9×11 растровых точек. Это дает возможность использовать компьютер как для графических приложений, так и для текстообработки.

Поскольку компьютер полностью совместим с IBM-PC, его можно использовать под управлением операционных систем PC-DOS, MS-DOS, CP/M-86, CON-CURRENT CP/M, UCSD-P, UNIX и другие.



Болгария, София, улица Чапаева, 51
Телефон: 74-61-51, Телекс: 022731 и 022732



УВАЖАЕМАЯ РЕДАКЦИЯ!

Я с удовольствием читаю Ваш журнал и нахожу в нем много полезного! Коснусь моих проблем. Я решил собрать микроЭВМ на основе КР580ИК80А, но меня не устроило ограничение объема ЗУ в 64К байт, поскольку задуманная мною микроЭВМ требует больших объемов данных: буфер цветного монитора (320×200 точек) — 16К байт; ПЗУ с монитором и наиболее часто используемыми программами — 16К байт; только качественный интерпретатор Бейсика с самой программой занимают больше 64К байт, а о компиляторе и говорить не приходится. Квазидиск в какой-то мере снял бы остроту проблемы, но не решил бы ее полностью.

Для расширения зоны ОЗУ я выбрал следующий путь: формировать сигналы на дополнительных адресных линиях (линии страниц P₀, P₁) при обращении к памяти

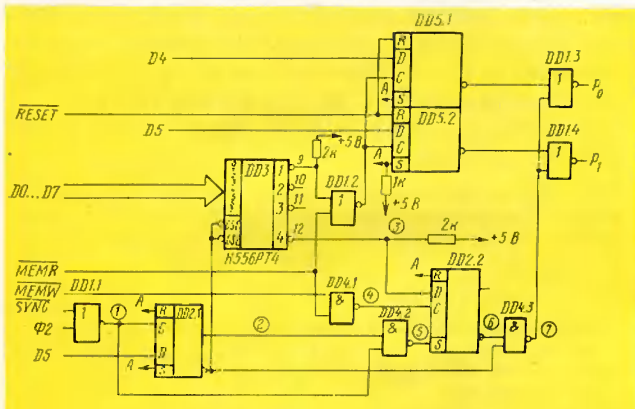


Рис. 1

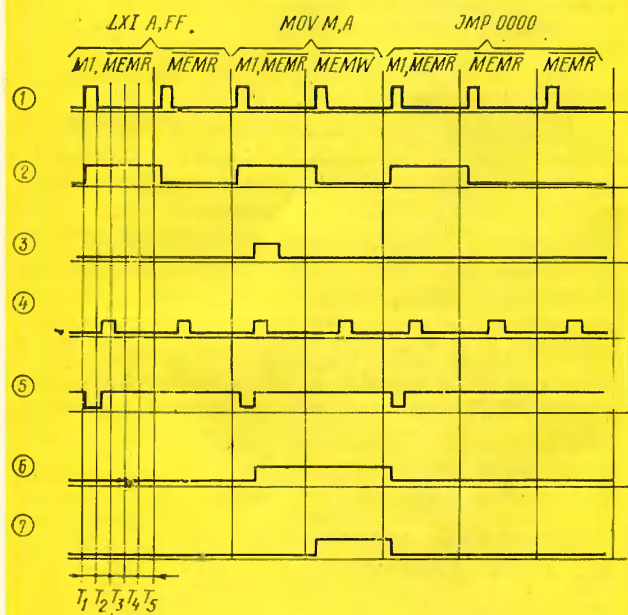


Рис. 2

(т. е. при выполнении команд типа MOV A, M; MOV M, C; ORA M и т. п.). Для загрузки регистра страниц, видимо, можно применять однобайтовые команды 08, 18, 28, 38, которые не используются процессором КР580ИК80А. Схема реализации этой функции показана на рис. 1, а эпюры напряжений в характерных точках — на рис. 2.

ПЗУ К556РТ4 должно быть запрограммировано так, чтобы на выводе 12 появлялся «Лог. 0» при выполнении команды обращения к памяти, а на выводе 9 — «Лог. 0» при выполнении любой из команд 08, 18, 28, 38.

Прошу ответить на вопросы:

1. Есть ли зерно нового в таком варианте формирования сигналов на дополнительных адресных линиях?

2. Допустимо ли использование однобайтовых команд 08, 18, 28, 38, или они не вполне аналогичны команде NOP?

3. Работоспособна ли схема на рис. 1?

4. Где можно прочитать об альтернативных путях решения этой проблемы?

И. Г. Муттик, инженер ГАИШ МГУ

Уважаемый Игорь Гарриевич!

1. Проблема использования памяти, объем которой превышает адресные возможности процессора, была весьма острой для 8- и 16-разрядных процессоров старых типов. Большинство процессоров нового поколения имеет встроены механизм расширения адресного пространства (до многих мегабайт). Стандартный способ расширения поля адресации — использование многоразрядных «регистров приписки». В Вашей схеме применяются эквивалентные им по функциям триггеры с выходом на дополнительные адресные линии.

2. Предложенный Вами способ управления дополнительными адресными линиями с помощью однобайтовых кодов некоторых неиспользуемых команд рекомендовать рискованно, так как выполнение их строго определенным образом не гарантируется изготовителями.

3. Если не считать этой тонкости, т. е. при условии, что эти команды будут выполнены так же, как команда NOP, предложенная Вами схема должна быть работоспособна. Надо только учитывать, что при любом изменении поля адресации после включения дополнительных разрядов адреса, процессору должна быть всегда доступна область памяти с активной в данный момент программой.

4. К сожалению, рекомендовать Вам литературу, освещающую вопросы расширения адресного пространства, трудно. В основном, это статьи в разных журналах.

Ю. И. Торгов, зав. отделом ВЦ АН СССР

ПЕРИФЕРИЙНЫЙ БЛОК ПАМЯТИ ДЛЯ РЕАЛИЗАЦИИ ВЕРСИИ ОС РАФОС

В. Д. Бахмацкий, В. Г. Белый, В. Л. Раппопорт, Л. А. Раппопорт

Для исследования возможностей ОС Рафос был спроектирован и изготовлен периферийный блок памяти (ПБП), который использовался в составе диалогового вычислительного комплекса. ПБП представляет собой ЗУ типа электронный «квазидиск»* и состоит из набора смежных модулей: пяти плат ППЗУ на основе микросхем К573РФ2 (см. фото), трех плат ОЗУ на основе микросхем К565РУ3 (К565РУ6) и платы управления.

ППЗУ хранит основные компоненты ОС: монитор, драйверы, программы копирования, верификации, контроля и модификации содержимого файлов, редактор текстов, трансляторы, библиотеки, компоновщик и др.

В ОЗУ размещаются файлы для обновления информации в процессе функционирования системы (например, файл SWAP.SYS) и разрабатываемые программы пользователя.

Плата управления обеспечивает связь блока с ЭВМ, имеющей канал «Электроника 60», содержит два адресных регистра-счетчика, 7-разрядный регистр младшей части и 16-разрядный регистр старшей части адреса (для доступа к адресному пространству 8М байт), 8-разрядный регистр данных и 16-разрядный регистр состояния. Плата включает программатор микросхем К573РФ2, аппаратно обеспечивая формирование временной диаграммы программирования, и резидентную часть ПЗУ (на двух микросхемах К556РТ5), хранящую начальный загрузчик операционной системы.



* Зеленко Г. В., Панов В. В., Попов С. Н. Электронный «квазидиск» для персональной ЭВМ//Микропроцессорные средства и системы, 1984 г., № 4, с. 79.

Для поддержки программатора создан драйвер FI.SYS. Предусмотрена возможность программирования отдельных микросхем К573РФ2 в съемном гнезде.

Технические характеристики ПБП

| | |
|--|-------------|
| Объем памяти ОЗУ, К байт | 128 |
| Объем памяти ППЗУ, К байт | 64 |
| Разрядность слова, бит | 8 |
| Среднее время обращения к ОЗУ, мкс | 3,2 |
| Время чтения ППЗУ, мкс, не более | 0,6 |
| Время записи ППЗУ, мс, не более | 60 |
| Напряжение питания, В (Гн) | 220(50) |
| Потребляемая мощность, Вт | 35 |
| Габаритные размеры, мм | 200×300×480 |
| Масса, кг | 5 |

В процессе работы программы реального времени, использующей дисковую память, время выполнения отдельных операций трудно учитывать из-за переменного места хранения файлов на диске. С применением ПБП время выполнения любой части программы легко поддается учету.

Длительность процедур в НГМД и НГМД, дополненном ПБП (в секундах)

| | | |
|---|------|------|
| Копирование (100 блоков) | 14,2 | 1,0 |
| Трансляция простой программы на ассемблере | 17,6 | 3,9 |
| Трансляция «большой» программы на ассемблере | 35,5 | 15,2 |
| Трансляция программы на Фортране | 130 | 15 |
| Компоновка простой программы | 12 | 1 |
| Компоновка «большой» программы | 750 | 150 |
| Максимальная скорость обмена | 10 | 70 |

Время загрузки оверлейного сегмента неоптимально для оператора.

Проведенная работа показала сравнительно невысокую трудоемкость адаптации операционной системы Рафос для записи в системный носитель ППЗУ. Использование ППЗУ увеличивает надежность комплекса за счет разгрузки дискового накопителя от системных запросов и уменьшает, в зависимости от типа программных операций, время работы в 10 и более раз. Стоимость комплектующих ПБП порядка 3000 рублей. Практическое использование ПБП показало, что он является высокоэффективным системным инструментом при разработке программ.

Адрес для справок: 340114, г. Донецк, ул. Розы Люксембург, д. 72, СКТБ ДФТИ АН УССР, отд. 51, т. 55-02-19, 55-71-31.

w, d — целые числа (w — число позиций, d — число цифр после точки); форматы — I (целый), F (с фиксированной точкой), E (с плавающей точкой), A (символьный), S (стандартный).

Функция CUR выдает текущую позицию курсора, а функция CLIN номер строки с курсором. Функция TEXT (x, y, [n]) выдает код символа в указанной позиции (n=0), цвет изображения и признак мигания (n=1), цвет фона (n=2).

Оператор CLS стирает изображения во всех позициях экрана, заполняя их фоновым цветом.

В графическом режиме можно отобразить точку, линию и более сложные объекты. Этот режим задается оператором SCREEN n. Обычно в ПК имеется несколько графических режимов, которые выбираются параметром n. Рекомендуются значения n=0 для графики низкого, n=1 среднего, n=2 высокого разрешения.

Оператор COLOR задает цвет изображения или номер палитры цветов (IBM PC, «Ириша»), цвет фона и рамки экрана. Поскольку размер раstra экрана зависит от аппаратных особенностей ПК, то возникают проблемы переноса программ на ПК с экранами другого разрешения. Для преодоления этих сложностей введен оператор задания мировых координат

WINDOW (x1, y1) — (x2, y2)

где (x1, y1) — новые координаты левой верхней точки экрана, а (x2, y2) — правой нижней точки. В последующих графических операторах параметры задаются в мировых координатах.

В следующих операторах координаты точек задаются абсолютно или относительно текущей позиции графического курсора. Во втором случае они предваряются словом STEP.

PSET/ PRESET (x, y) [, c] — отобразить точку цвета c. Если c опущен, то PRESET стирает точку.

LINE [(x1, y1)] — (x2, y2), [c] [, B [F]] — отобразить отрезок прямой

Если не задана первая точка, то она равна текущей; c — цвет линии, B — изображает прямоугольник на заданной диагонали, BF — вызывает закраску прямоугольника цветом c.

Функция POINT (x, y) выдает цвет указанной точки.

Функция POINT (n) выдает координаты графического курсора:

x (n=0), y (n=1).

CLS — заполнение экрана фоновым цветом

CIRCLE (x, y), r, [c], [(нач)], [(кон)], [(экс)] — изображение эллипса с центром в (x, y) и радиусом r, c — цвет линии. Для изображения дуги задают углы <нач> и <кон> в радианах, <экс> — отношение x-радиуса к y-радиусу

GET (x1, y1) — (x2, y2), v — содержимое экрана в заданном прямоугольнике считывается в массив

PUT (x1, y1) [(—(x2, y2)], v [, d] — прямоугольный образ отображается на заданную область экрана, d принимает значения PSET (замена содержимого экрана), NOT (замена с инверсией), AND, OR, XOR (логическая операция над старым и новым образами).

Оператор PUT используют для «оживления» графических объектов следующим образом. Выполнение оператора PUT с параметром XOR даст изображение объекта на экране. Повторное выполнение этого же оператора удалит объект без изменения остальной картины. Изменив координаты, снова выполняют оператор PUT. Повторение указанных шагов с временной задержкой между отображением и стиранием создаст иллюзию движения объекта. Меняя границы прямоугольной области в операторе PUT, можно изменить размеры отображаемого объекта. Если объект, созданный в графическом режиме низкого разрешения, отобразить с помощью оператора PUT в режиме более высокого разрешения и тех же мировых координат,

то он останется неизменным. Переход к более низкому разрешению лишь ухудшит качество изображения.

PAINT (x, y), [c1], [c2] — закраска замкнутой области цветом c1. Задается точка (x, y) внутри области и цвет границы c2.

DRAW <последовательность команд> — изображение сложных графических объектов с помощью набора команд. Команда имеет вид <буква> <константа> или <буква> =e;

Команды сдвига графического курсора с прорисовкой трассы: Up (вверх), Dn (вниз), Lp (влево), Rp (вправо), Fp (вниз — вправо), Gp (вниз — влево), Hp (вверх — влево), Ep (вверх — вправо), Mx, y (установка в указанную точку). Перед этими командами можно задать префиксы B (сдвиг без прорисовки), N (сдвиг и возврат в исходную позицию). Команда Sp задает длину элемента предыдущих команд. Команда Cc задает текущий цвет изображения, Pc1, c2 закрашивает область с границей цвета c2 цветом c1. Команда X=e; выполняет набор команд, заданных строкой символов e. В качестве параметров команд этой строки допускаются только константы.

Звуковые операторы. ПК может содержать один или более громкоговорителей. Для выдачи звука с частотой n1 (в Гц) и длительностью n2 (в 1/100 с) используется оператор SOUND [#n, n1, n2]

SOUND [#n, n1, n2]

Оператор PLAY озвучивает нотную запись:

PLAY (последовательность команд)

Команда имеет вид <буква> <целое> или <буква> =e; Команды от A до G задают ноты от «до» до «си» текущей октавы. Знаки + и # означают диез, а знак «минус» — бемоль. Команда On выбирает октаву, Np играет ноту с номером n, Lp задает длительность ноты, а точка после ноты увеличивает ее длительность наполовину. Команда Pp задает длительность паузы, Tp — темп в четвертях нот в минуту. X=e; выполняет последовательность команд, заданную строкой e. В качестве параметров этих команд используются только целые числа.

Внешние прерывания. Для обработки прерывания по ошибке при выполнении программы служат операторы и функции:

TRAP [m] — задание (отмена) режима программной обработки ошибок, ip — ссылка на подпрограмму обработки ошибок

RESUME [m] | NEXT — выход из подпрограммы обработки ошибок

ERROR p — мигания ошибки

Функция ERR — номер ошибки

Функция ERL — номер ошибочной строки.

Для обработки прерываний от внешних устройств предназначены операторы:

INTERRUPT p GOTO | GOSUB m — задание действия по прерыванию с номером p

INTERRUPT n, * или INTERRUPT * — отмена действий по одному или всем прерываниям

Для обработки прерывания от нажатой клавиши предназначены операторы:

KEY n GOTO | GOSUB m

KEY n, *

KEY *

Коды от «00»H до «FF»H соответствуют символам КОИ-8. Двухсимвольные параметры состоят из кода «00»H и номера функциональной клавиши («01»H... «0F»H). Число одновременно обрабатываемых клавиш в операторах KEY определяется реализацией языка.

К бытовому ПК можно подключить игровые пульты, имеющие кнопки и устройства изменения их состояния

(шар, колесо, ручка и т. п.). Поддерживается до четырех ракеток, каждая из которых имеет одну кнопку и изменяет свое значение в одномерной области, либо две мыши или джойстика с двумя кнопками каждая и с изменением состояния в двумерной области значений. Для обработки прерывания от кнопки игрового пульта предназначен оператор

STRIG n GOTO | GOSUB m,

где n — кнопка n -й ракетки или кнопка первого ($n=0, 1$) или второго ($n=2, 3$) джойстика (мыши). Операторы STRIG n , * и STRIG * отменяют действия при нажатии одной или всех кнопок. Функция STICK (n) выдает значение состояния n -й ракетки. Для двумерных игровых пультов по $n=0, 1$ выдаются значения x и y первого пульта, а по $n=2, 3$ — значения x и y второго пульта.

Стандартные функции: ABS (абсолютное значение), ATN (арктангенс), COS (косинус), SIN (синус), TAN (тангенс), INT (наибольшее целое, не превосходящее аргумента), SGN (знак числа), RND (случайное число), EXP (экспонента), LOG (логарифм), SQR (квадратный корень), FIX (округление до целого), DEG (перевод в градусы), RAD (перевод в градусы), VAL (перевод числа во внутреннее представление), NUM \square (перевод числа в символьное представление), CINT, CFLP, STR \square (преобразование к вещественному, целому, символьному типу), MID \square , LEFT \square , RIGHT \square (выделение подстроки), LEN (длина строки), REP \square (сцепление строк), POS (поиск подстроки), INC (код нажатой клавиши), FRE (свободная память), EOF (признак конца файла). Остальные функции были описаны выше.

Команды интерпретатора. Для удобства ввода команд они задаются одной буквой. В общем виде команд m — метка, n — целое число.

| | |
|--|--|
| ALST L C | — назначение устройства печати |
| CONT [m] | — продолжить выполнение |
| DEL [m] [-] [m], ... | — удалить строки программы |
| EDIT m | — редактировать строку программы |
| GEN [m] [n] | — автоматическая нумерация строк при вводе программы |
| INPUT « <i>имя файла</i> » | — ввод программы с внешнего носителя |
| JUMP | — конец работы |
| LIST [m] [-] [m], ... | — печать текста программы |
| OUTPUT « <i>имя файла</i> » [m] [-] [m], ... | — вывод программы на внешний носитель |
| PRINT <элементы печати> | — оператор PRINT |
| RUN [m] [-m] | — запуск программы |
| SCR | — удаление программы из памяти |
| TRACE [OFF] | — задание или отмена трассировки |
| UPDATE [m] [m] [n] | — перенумерация строк программы |
| \ <оператор> ... | — непосредственный режим выполнения операторов |

Язык Бейсик/Fs разделен на три уровня. Младший уровень языка полностью включается в более старший. Выбор уровня языка Бейсик/Fs определяется аппаратными возможностями ПК, его назначением. Уровень I языка Бейсик/Fs соответствует минимальному Бейсик-у, определенному Международным стандартом ISO [3]. Уровень II включает дополнительно работу с целыми числами, расширенный набор стандартных функций, развитые управляющие структуры (кроме SELECT), средства обработки ошибок, простейшие средства поддержки двухцветной графики (ввод-вывод

точки, работа в мировых координатах) и звука (оператор SOUND). Уровень III является полной версией языка Бейсик/Fs. Объемы интерпретаторов языка Бейсик/Fs I, II, III уровней соответственно 8К, 16К и 24...32К байт. Интерпретаторы уровней I и II обычно размещаются в постоянной памяти (ПЗУ) ПК, а интерпретатор уровня III в ПЗУ или на внешнем носителе (лента, диск). Во втором случае он загружается в память для выполнения средствами операционной системы или интерпретатора более низкого уровня (команда JUMP), размещенного в ПЗУ.

В операционных системах CP/M и МикроДОС имеются инструментальные средства (редактор текстов, перемещаемый макроассемблер, редактор связей, символический отладчик) для адаптации интерпретатора языка Бейсик/Fs на конкретный ПК с микропроцессором КР580ИК80А. Интерпретатор поставляется в виде набора логических и физических объектных модулей, причем для последних имеется исходный текст на языке ассемблера и инструкция по его адаптации. В физический модуль входят драйверы внешних устройств, сообщения интерпретатора, некоторые системные переменные.

Заключение. Язык программирования Бейсик/Fs в настоящее время реализуется для ПК на базе микропроцессора КР580ИК80А. Автор надеется, что полезные замечания и предложения читателей журнала позволят улучшить представленный проект единого языка программирования Бейсик для школьных и бытовых компьютеров. Для реализации языка Бейсик/Fs на ПК с процессорами, отличными от КР580ИК80А, могут быть предоставлены необходимые материалы и консультации.

Телефон для справок: 29-39-44 (г. Рига)

ЛИТЕРАТУРА

1. Трейстер Р. Персональный компьютер фирмы ИБМ.— М.: Мир, 1986.
2. MSX-BASIC. Programmins Manual. Sanyo Corp., 1985.
3. Minimal Basic. International Standard ISO 6373—1984.
4. Корчак А. Е. Интерпретатор языка Бейсик/Fs для микроЭВМ // Микропроцессорные средства и системы.— 1985.— № 3.— С. 21—24.
5. Корчак А. Е. Язык программирования Бейсик/Fs. Сер.: Методические материалы и документация по пакетам прикладных программ.— М.: МЦНТИ, 1986.— Вып. 44.
6. Ершов А. П. Алгоритмический язык в школьном курсе основ информатики и вычислительной техники // Микропроцессорные средства и системы.— 1985.— № 2.— С. 48—51.

Статья поступила 28 мая 1986 г.

УДК 681.32

А. В. Найденов, В. А. Романенков

ПРОГРАММАТОР ПЗУ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА К1-20»

При использовании небольших управляющих устройств на микропроцессорах часто необходимо записать программу объемом не более 2К байт в ПЗУ, чтобы вставить его затем в разрабатываемое устройство и проверить правильность выполнения программы в реальных условиях. Для этого целесообразно ввести блок программатора ПЗУ в состав микропроцессорного оценочного модуля, с помощью которого можно отладить такую программу. Кроме того, в некоторых применениях такой прибор будет использоваться

разработчиком только как автономный программатор ввиду больших функциональных возможностей.

Работа пользователя с программатором весьма проста и описана в табл. 1. Здесь адреса 6000H, 6010H, 6020H — это адреса пуска программ считывания содержимого КР556РТ5, К573РФ1 и К573РФ5 в буферное ОЗУ (начальный адрес которого 8000H, объем 4К байт). Адреса 6100H, 6110H, 6120H — это точки пуска программ прожига для тех же типов ППЗУ. Из-за высокой скорости выполнения режимов 2 (а и б), неувидимой для человеческого глаза, они не индицируются.

Устройство позволяет программировать ППЗУ следующих типов: КР556РТ5, К573РФ1, К573РФ5. Для упрощения программатора каждая ИС программируется в отдельной колодке. В колодке, предназначенной для занесения данных в ППЗУ типа К573РФ5, можно работать также с ППЗУ К573РФ2, совместимым по расположению выводов.

Программатор позволяет дополнительно работать с 8 типами ППЗУ. Для этого он дополняется набором кабелей, на одном конце ко-

торых установлена вилка, имеющая вид корпуса ИС ППЗУ, на другом — розетка, распаянная в соответствии с назначением выводов данного типа ППЗУ. Кабели подключаются к соответствующим розеткам: К155РЕ3, КР556РТ4 к розетке КР556РТ5; К573РФ13, К573РФ14 к розетке К573РФ1; К573РФ21, К573РФ22, К573РФ23, К573РФ24 к розетке К573РФ5. Возможны и другие способы распайки переходного кабеля для программирования дополнительных типов ППЗУ (например, использование сменных перемычек и т. п.).

Программатор имеет средства самодиагностики микроЭВМ «Электроника К1-20» и блока программатора. При проверке этой микроЭВМ они позволяют определить работоспособность микропроцессора, резидентного ОЗУ, содержащего стек и другие системные области, ППЗУ системного ПО, правильность работы пульта управления. Средства самодиагностики блока программирования проверяют работоспособность буферного ОЗУ: наличие замыканий в адресной части, возможное при изготовлении платы посредством пайки «волной», работу дешифратора адреса в кристалле ОЗУ, работу запоминающих ячеек кристалла ОЗУ. В соответствии с возможными неисправностями выдаются сообщения BAD или ADR.BAD.1. В случае полной работоспособности буферной памяти выдается сообщение GOOD.

Рассмотрим принципиальную схему блока программирования (рис. 1) и программное обеспечение прожига ППЗУ. Блок программирования состоит из следующих пяти узлов: передачи данных для ППЗУ серий К155 и КР556, программируемого источника питания (ПИП), трех формирователей напряжения высокого уровня, блока индикации и буферного ОЗУ.

Узел передачи данных собран на ИС DD1, DD2, DD5 и DD6. При программировании ППЗУ серий К155 и КР556 сигнал РТ5 имеет низкий уровень, и шинные формирователи DD1, DD2 подключаются к шине передачи данных. Сигнал ЧТ имеет высокий уровень, и записываемые данные через шинные формирователи и разрядные ключи на ИС DD5, DD6 поступают на вход ППЗУ. Рези-

Таблица 1

Работа пользователя с программатором

| Режим работы | Последовательность нажатия клавиш для выхода в данный режим | Индикация режима | Индикация результата выполнения режима |
|--|--|---|---|
| 1. Ввод данных с пульта управления | #, S, 8, 0, 0, 0 ... | Последовательно выводятся ячейки ОЗУ в шестнадцатеричном коде | Нет |
| 2. Ввод информации ППЗУ-эталона | #, G, 6, 0, 0, 0, CR (6, 0, 1, 0, CR) (6, 0, 2, 0 CR) | Нет | GOOD |
| 3. Ввод данных с микроЭВМ «Искра 226» | #, G, 6, 3, 0, 0, CR | J | GOOD |
| 4. а) Автоматическая проверка ППЗУ на возможность записи данной программы | #, G, 6, 1, 0, 0, CR (6, 1, 1, 0, CR) (6, 1, 2, 0, CR) | P | BAD1 (в случае невозможности прожига) |
| б) Программирование ППЗУ в) Автоматическая проверка правильности программирования | | — | — |
| 5. Самодиагностика программатора | 1) #, P, CR 2) #, B ... 3) #, G, 6, 2, 0, CR | Стандартный тест для микроЭВМ «Электроника К1-20» | GOOD или BAD (если прожиг ошибочен) » GOOD; либо BAD или BAD1 |

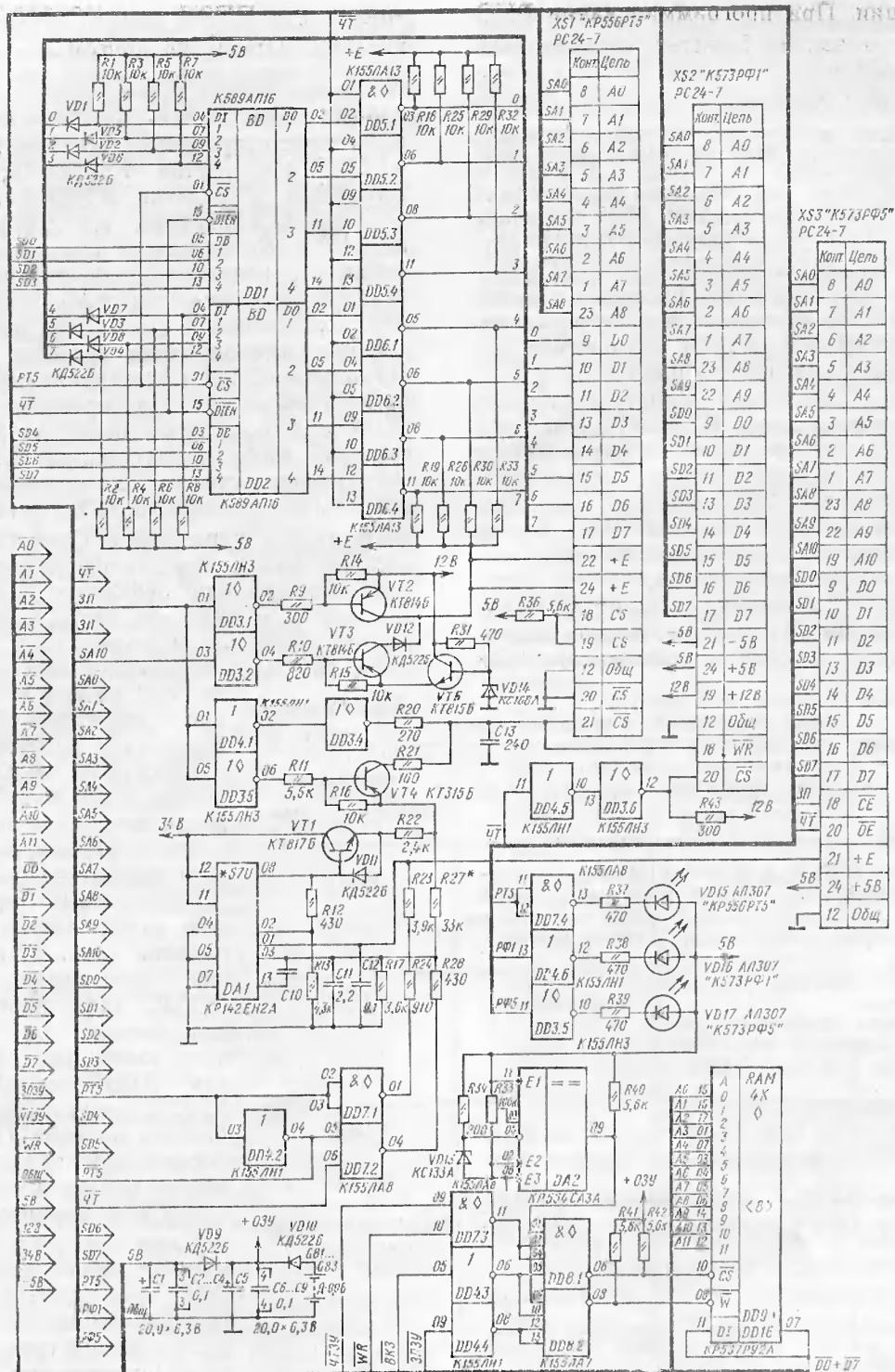


Рис. 1. Принципиальная схема программатора

сторы R18, R19, R25, R26, R29, R30, R32, R33 в режиме чтения являются нагрузочными, а в режиме записи задают ток прожига разрядов данных ППЗУ. В режиме чтения сигнал ЧТ имеет низкий уровень, поэтому транзисторы ключей DD5, DD6 закрыты и не влияют

на информацию, считываемую из ППЗУ через диоды VD1...VD8 и шинные формирователи на шину данных. Диоды открываются только при передаче низкого уровня, предохраняя входы шинных формирователей от сигналов уровня 12 В, поступающих на входы ППЗУ при запи-

си информации. При программировании БИС серии К573 сигнал РТ5 имеет высокий уровень, поэтому узел передачи данных от шины данных отключается.

Программируемый источник питания (ПИП) построен на основе стандартной схемы включения БИС стабилизатора типа КР142ЕН2Б. Программно-управляемый делитель на его выходе устанавливает все три напряжения (табл. 2), необходимые для программирования ППЗУ: 5, 15 и 25 В:

Таблица 2

Зависимость выходного напряжения ПИП от управляющих сигналов

| ЧТ | РТ5 | |
|----|------|------|
| | 1 | 0 |
| 1 | 15 В | 25 В |
| 0 | 15 В | 5 В |

Для защиты транзистора VT1, входящего в ПИП, от обратного напряжения включен диод VD11; конденсатор С11 устраняет выбросы напряжения при переключении ПИП с 25 на 5 В. Выбросы имеют значение (без С11) около 35 В и выводят из строя микросхемы К573РФ5. Применение ПИП существенно упростило схему программатора.

Узел формирователей напряжения подает на БИС ППЗУ высокоуровневые сигналы питания и программирования. Формирователи, собранные на DD3.1, DD3.2, VT2 и VT3, подают напряжение трех уровней питания (табл. 3) на входы +Е БИС ППЗУ серий К155, КР556 и нагрузочные резисторы.

Таблица 3

Зависимость выходного напряжения от управляющих сигналов

| А10 | ЭП | |
|-----|------|-----|
| | 1 | 0 |
| 1 | 12 В | 5 В |
| 0 | 12 В | 0 В |

Диод VD12 предохраняет транзистор VT3 от обратного напряжения при открытом транзисторе VT2, а параметрический стабилизатор VD13, VT5 с выходным напряжением около 6,1 В предназначен для компенсации падения напряжения на VD12 и VT3. Формирователи DD3.3, DD3.4, VT4 служат для подачи импульсов программирования с уровнем 15 или 25 В на ППЗУ серий К155, КР556 и К573РФ1 соответственно. Здесь конденсатор С13 служит для ограничения крутизны фронта импульса программирования. Длительность фронта импульса должна быть не меньше времени нарастания фронта импульса программи-

рования для ППЗУ типа КР556РТ5 и не больше времени нарастания фронта импульса программирования для ППЗУ типа К573РФ1; таким образом, она выбрана равной 400 нс.

Третий формирователь, собранный на элементах DD3.6, DD4.5, подает на вход CS/PGM ППЗУ типа К573РФ1 напряжение +12 В при записи и нулевой уровень при чтении.

Для индикации типа ППЗУ, с которым работает пользователь, предназначены светодиоды VD15...VD17.

Буферное ОЗУ (4К байт), используемое в блоке программатора, построено на БИС КР537РУ2А (4К×1). Такого объема вполне достаточно для работы с ППЗУ емкостью до 2К байт. Использование этого типа БИС ОЗУ позволило отказаться от канальных приемопередатчиков при построении блока памяти. Так как эти БИС ОЗУ изготовлены по к-МОП технологии и блок памяти, состоящий из 8 микросхем DD9...DD16, потребляет в статическом режиме ток ≤400 мкА, появляется возможность сделать такое ОЗУ энергонезависимым, питая его от аккумуляторов типа Д-0,06Д (U_{ном}=1,25 В).

Применение энергонезависимого буферного ОЗУ облегчает работу пользователя: теперь он получает возможность подготавливать информацию для занесения в ППЗУ в течение нескольких дней, а также выключать программатор в перерывах. Небольшие затраты аппаратуры (всего 16 корпусов ИС) и широкие функциональные возможности программатора (программирование 11 типов ППЗУ) обусловлены применением микроЭВМ «Электроника К1-20» с соответствующим программным обеспечением (ПО) программатора.

Программное обеспечение блока программатора (400 байт) записано в БИС ППЗУ К573РФ5, установленную в одну из резервных колодок на плате микроЭВМ «Электроника К1-20» с начальным адресом 6000Н. ПО программатора состоит из трех «параллельных ветвей» программ работы с ППЗУ типов КР556РТ5, К573РФ1 и К573РФ2. Эти «ветви программ» максимально унифицированы и имеют общие подпрограммы: ввода с ППЗУ-эталопа, проверки на возможность прожига, проверки правильности программирования, индикации режимов прожига и результата выполнения режима.

Отличаются лишь программы, реализующие алгоритмы программирования микросхем трех указанных выше типов. Сводные диаграммы записи и чтения ППЗУ трех типов приведены на рис. 2.

Пример. На программах записи (чтения) данных в ППЗУ КР556РТ5 рассмотрим особенности ПО и его взаимодействие с аппаратурой блока программирования, так как про-

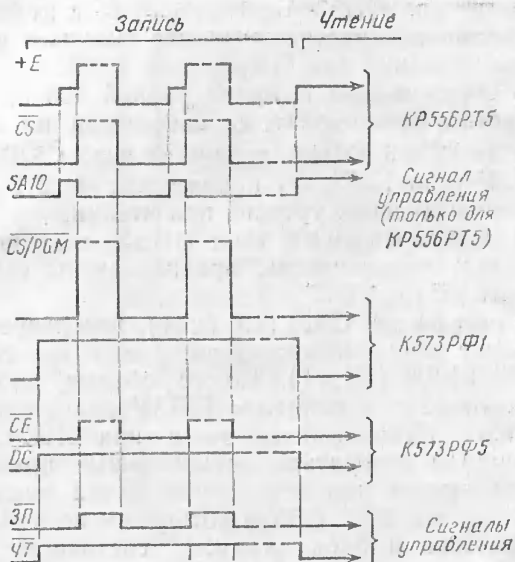


Рис. 2. Временные диаграммы сигналов управления (масштаб времени не выдержан)

грамма работы с этим типом ППЗУ имеет наиболее сложный алгоритм, реализуемый практически всей аппаратурой программатора.

Программа записи информации в ППЗУ КР556РТ5 (табл. 4) обеспечивает упрощенный алгоритм с фиксированной длительностью импульса прожига. Такое упрощение, существенно сокращающее объем ПО, целесообразно при программировании небольших партий ППЗУ. Работу программатора обеспечивает программируемый параллельный интерфейс (ППИ), установленный на плате микроЭВМ «Электроника К1-20» и работающий в режиме 0. При записи ППЗУ все порты настроены на вывод. При копировании ППЗУ, проверке на возможность и правильность прожига порты В и С настроены на вывод, порт А — на ввод. На выходе ППИ стоят шинные формирователи (ШФ) К582АП26 (направленные передачи устанавливается с помощью перемычек). Вход \overline{DIEN} ШФ портов В и С подключают к общему проводу, а вход \overline{DIEN} ШФ порта А — к разряду С6 порта С.

Порт А (адрес 0F4H) используется для передачи данных, порт В (адрес 0F5H) — для передачи младших разрядов адреса, назначенные разряды порта С приведено в табл. 5.

Таблица 5

| Назначение разрядов порта С | | | | | | | |
|-----------------------------|----|-----|-----|-----|-----|----|----|
| Порт С (адрес 0F6H) | | | | | | | |
| C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| As | A9 | A10 | PT5 | PФ5 | PФ1 | ЧТ | ЗП |

В исходном состоянии в БИС КР556РТ5 записаны «Лог. 1». Для записи «Лог. 0» в ка-

```

ПРОГРАММА ЗАПИСИ КР556РТ5
T5: EQU 01FFH ; ОБЪЕМ ПЗУ КР556РТ5
RAM: EQU 0000H ; НАЧАЛО БУФЕРА ДАННЫХ
CS: EQU 0F7H ; ПИИ: P5C
CONTR: EQU 0F4H ; ПИИ: P5C
ADR: EQU 0F5H ; ПИИ: P5C
DATA: EQU 0F4H ; ПОРТ А
SIGN: EQU 0E4H ; ПИИ: ПОРТ С
NUMBER: EQU 0B6H ; ПОРТ В
N1: EQU 07 ; ЧИСЛО ЦИКЛОВ ПРОГРАММИРОВАНИЯ
R0: EQU 0F2H ; РЕЖИМ: ЧТЕНИЕ КР556РТ5
NR0: EQU 0B6H ; ЗАПИСЬ: E=0B, CS=0E
NR1: EQU 0B2H ; ЗАПИСЬ: E=0B, CS=0E
NR2: EQU 034H ; ЗАПИСЬ: E=12B, CS=15B
T4: EQU 0F0H ; УСТАНОВ. ПАУЗ: 15 НС
T1: EQU 04 ; ВРЕМЯ ИМПУЛЬСА ПРОГРАММИРОВАНИЯ-0.3 МС
T2: EQU 020H ; ВРЕМЯ ПАУЗЫ- 2.7 МС

FRIS: ORG 6100H ; РЕЖИМ - ЧТЕНИЕ КР556РТ5
CALL TEST ; ПРОВЕРКА НА ВОЗМОЖНОСТЬ ПРОЖИГА
MVI B,WRD ; РЕЖИМ - ЗАПИСЬ КР556РТ5, E=0B, CS=03
CALL ROMIR ; УСТАНОВ. НАЧАЛЬНОГО АДРЕСА ПЗУ
LXI D,T5 ; УСТАНОВ. СЧЕТЧИКА КОЛИЧЕСТВА ЦИКЛОВ
MVI C,NR0 ; УСТАНОВ. МАСКИ НА НАЧАЛО
NOV ; ЧИТАТЬ ДАННЫЕ ИЗ ОЗУ
ORA C ; ИНВЕРТИРОВАТЬ ДАННЫЕ
ANA M ; ПРОВЕРКА НА ВОЗМОЖНОСТЬ ПРОЖИГА СИТА
JZ SHIF ; ЕСЛИ ПРОЖИГ НЕ НАБЛЮДАЕТСЯ, ПЕРЕИТИ К ПРОЖИГУ СЛЕДУЮЩЕГО БИТА
OUT DATA ; ВЫВЕСТИ ДАННЫЕ
OUT ADR ; ВЫВЕСТИ АДРЕСЫ БАЙТ АДРЕСА
MVI A,NR1 ; РЕЖИМ: E=5B, CS=0B
ORF D
OUT CONTR ; РЕЖИМ: E=12B, CS=15B
ORA 0
OUT CONTR ; ЗАДЕРЖКА 300 НКС
MVI A,T1 ; ЗАДЕРЖКА
TIME ; РЕЖИМ: E=0B, CS=0E
ORA 0
OUT CONTR ; ЗАДЕРЖКА 27 НС
CALL ROMV ; СДВИГ МАСКИ
MVI A,B ; ЕСЛИ МАСКА СДВИНУТА НЕ ДО КОНЦА,
L1 ; ПРОДОЛЖИТЬ ЗАПИСЬ БАЙТ
ORC ; ЕСЛИ ЧИСЛО ЦИКЛОВ ПРОЖИГА МЕНЬШЕ
JNZ L2 ; ОБЪЕДИН. ПРОЖИГА МЕНЬШЕ
INX H ; ПЕРЕИТИ К СЛЕДУЮЩИМ АДРЕСАМ ОЗУ
DCR D ; И ПЗУ
MOV A,D ; ЕСЛИ АДРЕС НЕ ПОСЛЕДНИЙ, ПРОДОЛЖИТЬ
CPL 0F4H ; ПРОЖИГ
JNZ L3 ; ПРОЖИГ
CALL RDT5 ; РЕЖИМ - ЧТЕНИЕ КР556РТ5

CALL EXAM ; ПРОВЕРКА НА ПРАВИЛЬНОСТЬ ПРОЖИГА

RDT5: PUL ; РЕЖИМ - ЧТЕНИЕ КР556РТ5
LXI D,T5 ; УСТАНОВИТЬ НАЧАЛЬНУЮ АДРЕС ПЗУ
MVI B,RD ; РЕЖИМ - ЧТЕНИЕ КР556РТ5
CALL ROMRD
RET

ROMRD: PUL ; РЕЖИМ - ЧТЕНИЕ ПЗУ
MVI A,0FH ; НАСТРОИТЬ ППИ: ПОРТ А - ВВОД,
OUT CS ; ПОРТЫ В И С - ВЫВОД, РЕЖИМ 0
LXI H,RAI ; НАСТРОИТЬ ППИ
MOV A,B ; НАСТРОИТЬ ППИ
OUT CONTR ; ЗАДЕРЖКА 15 НС
MVI A,0F0H ; ЗАДЕРЖКА 65 НКС (1 ЦИКЛ)
XTIME ; ЗАДЕРЖКА
XTIME ; ЗАДЕРЖКА
XTIME ; ЗАДЕРЖКА
DCR A
JNZ TIME
RET

ROMIR: PUL ; РЕЖИМ - ЗАПИСЬ ПЗУ
MVI A,0B6H ; ВЫВЕСТИ СИМВОЛ "P"
OUT SIGN ; НА ДИСПЛЕЕ ПУЛТА
MVI A,0B6H
OUT NUMBER
MVI A,0B6H ; НАСТРОИТЬ ППИ: ПОРТЫ А, В, С -
JMP SET ; ВВОД, РЕЖИМ=0

TEST: PUL ; ПРОВЕРКА НА ВОЗМОЖНОСТЬ ПРОЖИГА
CALL ROMRD ; ЧИТАТЬ ПЗУ
XRA M ; ПРОВЕРКА НА ВОЗМОЖНОСТЬ
AND M ; ПРОЖИГА
JNZ BAD1 ; ЕСЛИ ПРОЖИГ НЕВОЗМОЖЕН,
; ИНДИКАЦИЯ "BAD"
INX H ; УВЕЛИЧИТЬ НА 1 АДРЕС ОЗУ
DCR D ; И ПЗУ
MOV A,D ; ЕСЛИ АДРЕС НЕ ПОСЛЕДНИЙ,
CPL 0F4H ; ПРОДОЛЖИТЬ ПРОВЕРКУ
JNZ TEST ; ПРОДОЛЖИТЬ ПРОВЕРКУ
RET

EXAM: PUL ; ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОЖИГА
CALL ROMRD ; ЧИТАТЬ ПЗУ
MVI A,0 ; ПРОВЕРКА ПРАВИЛЬНОСТИ ПРОЖИГА
AND M ; ПРИ ОШИБКЕ-ИНДИКАЦИЯ "BAD"
INX H ; УВЕЛИЧИТЬ НА 1 АДРЕС ОЗУ И ПЗУ
DCR D
MOV A,D ; ЕСЛИ АДРЕС НЕ ПОСЛЕДНИЙ,
CPL 0F4H ; ПРОДОЛЖИТЬ ПРОВЕРКУ
JNZ TEST ; ПРОДОЛЖИТЬ ПРОВЕРКУ
JMP ROMRD ; ИНДИКАЦИЯ "GOOD"

ROMV: PUL ; ЧТЕНИЯ ПЗУ
MVI A,B ; ВЫВЕСТИ АДРЕСА A6, A9 И СИГНАЛ
ORA D ; УПРАВЛЕНИЯ
CONTR ;
OUT ;
MVI A,E ; ВЫВЕСТИ АДРЕСА A6-A7
MVI A,R ;
OUT ;
IN ;
ORA ;
MVI A,C ; ИНВЕРТИРОВАТЬ ДАННЫЕ
RET

```

кую-либо ячейку микросхемы необходимо подать на входы питания «Е» и нагрузочные резисторы R18, R19, R25, R29 сигнал с формирователя, построенного на VT2 и VT3.

В каждый момент времени на ключи должна быть подана информация об одном бите, на всех остальных ключах низкий уровень должен шунтировать остальные биты данных ППЗУ. Чтобы сократить время прожига, программа записывает информацию лишь в те биты, в которые нужно записать уровень «Лог. 0». Программируют ППЗУ импульсы напряжения амплитудой 15 В и длительностью 300 мкс, следующие с паузами 2,7 мс. По каждому биту информация записывается 7 раз. Число циклов записи, а также длительность импульса программирования и паузы задаются программно. Практический опыт программирования БИС КР556РТ5 показал, что заданные параметры прожига удовлетворительны для 80...90 % ППЗУ, а остальные кристаллы, как правило, не программируются и при повторной записи.

Программа чтения информации в ППЗУ КР556РТ5 (табл. 6) считывает ППЗУ с начального адреса и записывает его содержимое в буферное ОЗУ, начиная с адреса 8000H.

ТАБЛИЦА 6
ПРОГРАММА ЧТЕНИЯ КР556РТ5

| | | |
|------|-----------------|---------------------------------|
| ORG | 8000H | |
| CALL | RDT5 | ПРЕИМ ЧТЕНИЯ КР556РТ5 |
| CALL | COPY | КОПИРОВАТЬ ПЗУ |
| П/Л | КОПИРОВАНИЯ ПЗУ | |
| CALL | READ | ЧИТАТЬ ПЗУ |
| MOV | H, A | ЗАПИСАТЬ В ОЗУ |
| EXX | B | УВЕЛИЧИТЬ НА 1 АДРЕСА ОЗУ И ПЗУ |
| DCX | B | |
| MOV | A, B | |
| CPI | OFFH | ЕСЛИ АДРЕС НЕ ПОСЛЕДНИЙ, |
| | | КОПИРОВАТЬ ДАЛЬШЕ |
| JMP | LOOP | ИНДИКАЦИЯ "LOOP" |

Программы работы с ППЗУ серии К573 выполнены аналогично.

Телефон для справок о приобретении документации: 1-11-87 (Смоленск). Запросы и сведения о потребности в программах на 1986—1990 гг. направлять по адресу: 214000, Смоленск, ул. Шевченко, 97, СКТБ СПУ.

Статья поступила 15 января 1986 г.

УДК 681.3.06

И. К. Половинкин

КРОСС-СИСТЕМА ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА ДЗ-28» НА БАЗЕ ЭВМ «ИСКРА 226»

Системное программное обеспечение микроЭВМ «Электроника ДЗ-28» рассчитано на трудоемкое программирование в машинных кодах, требующее высокой квалификации программиста и значительного времени для отладки. Представленный с микроЭВМ Бейсик-интерпретатор несколько ускоряет разработку программы, но существенно снижает возможности системы. Он занимает значительный объем оперативной памяти, имеет невысокое быстродействие и не позволяет вести обработку символьных данных. Кроме того, для микроЭВМ «Электроника ДЗ-28» разработана кросс-система МИКРО-ИНФ, реализованная на ЕС ЭВМ [1]. Однако в этой системе отладку программы можно полностью выпол-

нить лишь на микроЭВМ. После выявления ошибок и коррекции программы требуется новая трансляция и передача на микроЭВМ.

Кросс-система подготовки программ на персональной ЭВМ «Искра 226» обеспечивает возможности ввода, полной отладки и компиляции программ в диалоговом режиме. Передача объектной программы на микроЭВМ «Электроника ДЗ-28» осуществляется через устройство обмена, работающее в режиме ИРПР и подключаемое к БИФ «Искра 015-82» [2]. В ПО системы входят:

интерпретатор Бейсик-2 (версия от 30.09.84), приемлемый на микроЭВМ «Искра 226»;

кросс-компилятор, разработанный для данной системы; драйверы устройства межмашинного обмена.

Технология программирования в кросс-системе включает следующие операции:

ввод программы с клавиатуры с визуальным контролем с помощью блока отображения символьной и графической информации (БОСИ);

полная отладка программы в режиме интерпретации; компиляция программы с записью оттранслированной программы в архив;

передача объектной программы в микроЭВМ «Электроника ДЗ-28».

Для хранения промежуточных результатов на всех операциях используются гибкие магнитные диски. Значительную часть функций в системе выполняет интерпретатор. На этапе ввода программы осуществляется синтаксический контроль. Предусмотрены возможности ввода основных операторов и функций Бейсика с помощью одной клавиши, исправления ошибок и редактирования строк без повторного ввода. При выполнении отлаживаемой программы на БОСИ отображаются возникающие ошибки. Обеспечены возможности пооператорного выполнения программы, индикации промежуточных и конечных результатов счета. В качестве исходного модуля для кросс-компилятора используется Бейсик-программа. Более удобен для этой цели промежуточный код, являющийся результатом обработки Бейсик-программы интерпретатором и обладающий следующими особенностями:

содержит таблицу переменных; элементы языка имеют идентификаторы, удобные для опознания;

каждый оператор и каждый номер строки снабжены информацией о длине оператора (строки). Такая структура программы облегчает выполнение проходов кросс-компилятором.

Интерпретатор располагает развитыми средствами обработки информации байтового формата, что позволяет использовать Бейсик в качестве языка программирования самого кросс-компилятора. На Бейсике реализована также программа, обслуживающая межмашинный обмен со стороны ПЭВМ «Искра 226». Кросс-компилятор занимает 28К байт оперативной памяти микроЭВМ «Искра 226». Наряду с кросс-компилятором разработаны подпрограммы в машинных кодах «Электроника ДЗ-28» (объем 3,3К байта), реализующие операторы и отдельные формы операторов языка Бейсик. Эти подпрограммы включаются в объектную программу по мере необходимости.

Опыт эксплуатации кросс-компилятора показал, что быстродействие оттранслированных программ в 5...8 раз выше быстродействия программ на Бейсике микроЭВМ «Электроника ДЗ-28» (вариант 3А). Объем оттранслированных программ в 1,1...1,5 раза больше, чем Бейсик-программ, однако отсутствие интерпретатора дает значительный выигрыш оперативной памяти. Экономия памяти может быть получена также за счет упаковки цифровых данных в символьные. Имеется возможность включения в программу кодовых вставок произвольной длины, которые позволяют с минимальными ограничениями использовать возможности системы команд микроЭВМ «Электроника ДЗ-28».

Адрес для запроса дополнительной информации:

ЛИТЕРАТУРА

1. Б. И. Гехман, И. А. Равкин, В. Л. Темов. Кросс-система для микроЭВМ «Электроника ДЗ-28». — Электронная техника. Сер. 9. Экономика и системы управления. Вып. 3 (52). — М.: ЦНИИ «Электроника», 1984.
2. С. Н. Абрамович, В. В. Бойко, В. П. Бутрин и др. Профессиональные персональные ЭВМ «Искра 226». — Микропроцессорные средства и системы, 1985, № 2, с. 29—36.

Статья поступила 29 августа 1985 г.

УДК 681.322

В. И. Жихарев

ПРОГРАММАТОР НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА ДЗ-28»

Программатор вместе с микроЭВМ «Электроника ДЗ-28» и алфавитно-цифровым дисплеем позволяет: редактировать информацию для записи в ПЗУ; отображать на экране дисплея записываемую информацию, содержащее ПЗУ в удобной для восприятия и понимания форме; автоматизировать программирование и контроль правильности ввода информации в ПЗУ.

Особенность программатора: при небольших аппаратных затратах с его помощью решаются достаточно сложные задачи программирования ПЗУ. Для

этого в память микроЭВМ вводится программа управления программатором.

Работа программатора в режиме записи (рис. 1). С помощью дешифратора D3, состояние которого меняется шиной управления микроЭВМ, формируются четыре stroba. Тремя из них информация с шины вывода микроЭВМ записывается в три регистра: адресный (микросхемы D1, D2), управляющий (D6), информационный (D5). Четвертый strob служит для запуска генератора программирующих импульсов. Генератор тактовых импульсов D14, V12 заполняет счетчик D12, D13 за 0,4 с. За это время цифро-аналоговый преобразователь D15 и операционный усилитель D18 формируют линейно-возрастающее пилообразное напряжение, пропорциональное по величине числу тактовых импульсов или выходному коду счетчика. Уровень и амплитуда напряжения устанавливаются потенциометрами R21, R22 соответственно. Опротон V1, питаемый этим напряжением, осуществляет широтно-импульсную модуляцию генератора V2, D10. Длительность импульсов этого генератора, начиная от 1 мкс, линейно возрастает (в течение 0,4 с) до 8 мкс при постоянной скважности [2]. Эти импульсы подаются на управляемые ключи D7, D9, позволяющие в течение цикла (0,4 с) программировать только один бит. Пять усилителей мощности V3, V7, V4, V8, V5, V9, V6, V10, V11, V13 обеспечивают требуемые для K541PT1 величины напряжений и токов в режимах записи, подаваемые на соответствующие ее выходы. Одновременно с подачей импульсов на один из входов Q1, Q2, Q3, Q4 микросхемы K541PT1 такая же последовательность импульсов подается на ее 14-й вывод; если бит программировать не надо, то подавать ее на 14-й вывод не нужно. Эта процедура проводится

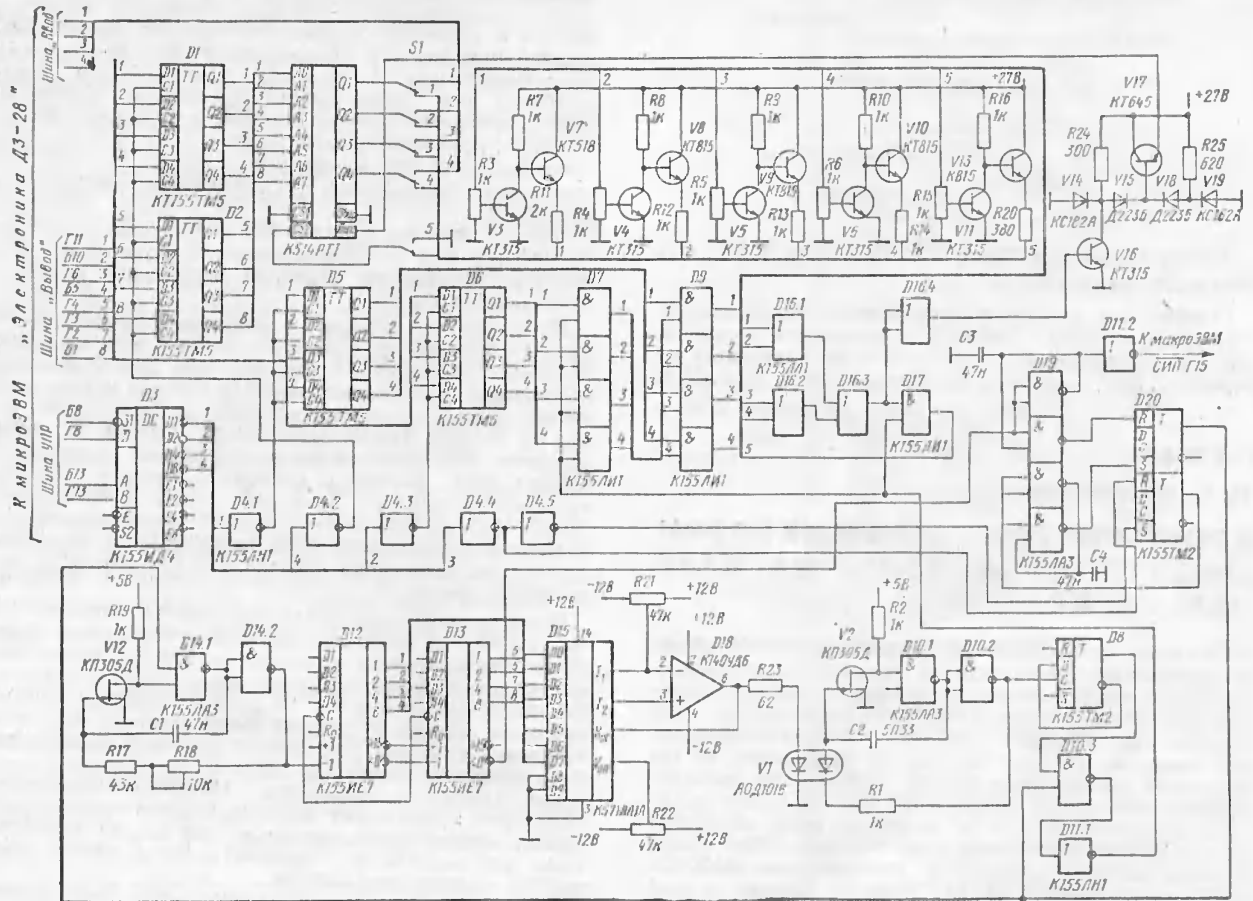


Рис. 1. Принципиальная схема программатора

с помощью микросхем D16, D17. Они суммируют X1, X2, X3, X4 (входные сигналы микросхем, упомянутых выше, соответствующие определенным разрядам записываемого слова), затем умножают их сумму на X5 — программирующую последовательность импульсов. Созданный таким образом выходной сигнал у подается через усилитель V11, V13 на 14-й вывод микросхемы K541PT1. С помощью транзисторов V16, V17, стабилизаторов V14, V19 и диодов V15, V18 напряжение на 16-м выводе микросхемы K541PT1 в момент программирования бита с +5 В увеличивается до +8 В.

Формирователи D19, триггеры D20 синхронизируют работу программатора.

По окончании цикла программирования формирователь D19 выдает импульс СИП, который через элемент D11.2 поступает на шину ввода микроЭВМ (гнездо П15). Импульсом СИП программатор извещает микроЭВМ об окончании цикла программирования и готовности принять от нее следующую порцию информации: адрес записываемого слова, само слово и 4-разрядный код для управляющего регистра D6.

Программа обслуживания программатора состоит из трех подпрограмм.

Первая расположена с нулевого шага по 120-й. С ее помощью на экран дисплея из области ОЗУ микроЭВМ с начальным адресом, указанным в регистре микроЭВМ R0, выводятся 256 4-разрядных слов, представленных нулями и единицами.

Программа обслуживания программатора

| | |
|-----------------|--|
| 0 MOVH X, R0 | Установка начального адреса ОЗУ |
| 2 CLR R4 | Обнуление регистра R4 |
| 4 CLR R10 | Обнуление регистра R10 |
| 6 MOV#0300, S | Запись кода единицы в регистр S0 |
| 8 MOV#0301, S1 | Запись кода единицы в регистр S1 |
| 10 MOV#0200, S6 | Запись кода пробела в регистр S6 |
| 12 MOV#0010, S7 | Запись кода перевода строки в регистр S7 |
| 14 CLR R1 | Обнуление регистра R1 |
| 16 DIG 1 | |
| 17 DIG 5 | |
| 18 MOVH X, R2 | Запись числа 15 в регистр R2 |
| 20 DIG 1 | |
| 21 DIG 5 | |
| 22 MOVH X, R3 | Запись числа 15 в регистр R3 |
| 24 MOV#1507, S2 | |
| 26 MOV#1515, S3 | |
| 28 OUTOWS | Обращение к дисплею |
| 30 MARK 1 | Установка метки 1 |
| 32 BB1C 4, @R0 | Если четвертый бит в байте, расположенном по адресу R0, будет нулевым, то на экран дисплея выводится ноль. В противном случае — единица |
| 34 BR 42 | |
| 36 MOV S0, @R10 | |
| 38 OUTO 1501 | |
| 40 BR 46 | |
| 42 MOV S1, @R10 | |
| 44 OUTO 1501 | |
| 46 BB1C 5, @R0 | Если пятый бит в байте, расположенном по адресу R0, будет нулевым, то на экране дисплея изображается ноль. В противном случае — единица |
| 48 BR 56 | |
| 50 MOV S0, @R10 | |
| 52 OUTO 1502 | |
| 54 BR 60 | |
| 56 MOV S1, @R10 | |
| 58 OUTO 1502 | |
| 60 BB1C 6, @R0 | Если шестой бит в байте, расположенном по адресу R0, будет нулевым, то на экране дисплея изображается ноль. В противном случае — единица |
| 62 BR 70 | |
| 64 MOV S0, @R10 | |
| 66 OUTO 1501 | |
| 68 BR 74 | |
| 70 MOV S1, @R10 | |

72 OUTO 1501
74 BB1C 07, @E0
76 BR 84
78 MOV S0, @R10
80 OUTO 1501
82 BR 88
84 MOV S1, @R10

88 MOV S6, @R10

90 OUTO 1501

92 ADD#1, R0

94 ADD#1, R1

96 BSA R1, R2

98 JMM 1

100 MOV S7, @R10

102 OUTO 1501

104 CLR R1

106 ADD#1, R4

108 BSA R3, R4

110 JMM 01

112 OUTO 1501

114 OUTO 1501

116 OUTO 1501

118 OUTO 1501

120 STOP

Если седьмой бит в байте, расположенном по адресу R0, будет нулевым, то на экран дисплея выводится ноль. В противном случае — единица

Запись в ОЗУ по адресу R10 кода единицы
Запись в ОЗУ по адресу R10 кода пробела

Вывод на экран изображения пробела

Увеличение на единицу содержимого регистра R0

Увеличение на единицу содержимого регистра R1

Если содержимое регистров R1, R2 не одинаково, то переход к метке 1, если равны, то перевод строки на экране дисплея

Обнуление регистра R1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

Увеличение на единицу содержимого регистра R4. Если содержимое регистров R3, R4 одинаково, то на экране дисплея четыре раза подряд осуществляется перевод строки. Если не равны, то переход к метке 1

154 STOP

Третья подпрограмма автоматически проверяет правильность записи 256 тетрад в ПЗУ. 256 тетрад, подлежащих записи, побитно сравниваются по логическому закону «Исключающее ИЛИ» с 256 тетрадами, считанными из ПЗУ по окончании записи. Результат проверки выводится на экран дисплея первой подпрограммой из области памяти микроЭВМ с начальным адресом, находящимся в регистре R4. При правильной записи на экране появятся 256 тетрад, все биты которых будут нулями. Единичные биты укажут место ошибки программирования.

Подпрограмма контроля правильности ввода информации в ПЗУ

| | |
|-----------------|---|
| 155 MARK 5 | |
| 157 CLR R11 | |
| 159 CLR R12 | |
| 161 CLR R2 | |
| 163 CLR X | |
| 164 STOP | |
| 165 MOVH X, R00 | |
| 167 CLR X | |
| 168 STOP | |
| 169 MOVH X, R01 | |
| 171 CLR X | |
| 172 STOP | |
| 173 MOVH X, R4 | |
| 175 DIG2 | |
| 176 DIG5 | |
| 177 DIG6 | |
| 178 MOVH X, R3 | |
| 180 MARK 6 | |
| 182 MOV @R0, S6 | Пересылка байта из ОЗУ по адресу R0 в S6 |
| 184 MOV @R1, S8 | Пересылка байта из ОЗУ по адресу R1 в S8 |
| 186 XOR S6, S8 | Сравнение S6, S8 по закону «Исключающее ИЛИ» |
| 188 MOV S8, @R4 | Пересылка результата сравнения по адресу R4 |
| 190 ADD#1, R0 | |
| 192 ADD#1, R1 | |
| 194 ADD#1, R4 | Увеличение содержимого регистров R0, R1, R4, R2 на единицу |
| 196 ADD#1, R2 | |
| 198 BSA R2, R3 | Если содержимое регистров R2, R3 одинаково, то останов подпрограммы. Если нет, то переход к метке 6 |
| 200 JMM 6 | |
| 202 STOP | |
| 203 END | Конец |

Программа считывания информации из ПЗУ позволяет размещать ее в любом месте оперативной памяти микроЭВМ, начальный адрес которой записан в регистре R2.

Программа считывания информации из ПЗУ и размещения ее в ОЗУ микроЭВМ «Электроника Д3-25» с начального адреса R2

| | |
|-----------------|--|
| 0 DIG2 | |
| 1 DIG5 | |
| 2 DIG6 | |
| 3 MOVH X, R4 | Запись числа 256 в регистр R4 |
| 5 CLR R11 | Обнуление регистра R11 |
| 7 CLR R1 | Обнуление регистра R1 |
| 9 MARK 7 | Установка метки 7 |
| 11 MOV#0400, S2 | |
| 13 MOV S7, S3 | |
| 15 OUTOWS | |
| 17 INPOWC | Принем в регистр S3 слова ПЗУ |
| 19 MOV S3, @R2 | Пересылка слова ПЗУ по адресу R2 в ОЗУ |
| 21 ADD#1, R11 | Увеличение на единицу содержимого регистра R11 |

| | |
|---------------|---|
| 23 ADD#1, R2 | Увеличение на единицу содержимого регистра R2 |
| 25 ADD#1, R1 | Увеличение на единицу содержимого регистра R1 |
| 27 BSA R4, R1 | Если содержимое регистров R4, R1 одинаково, то конец программы. Если нет, то переход к метке 7. |
| 29 JMM 7 | |
| 31 STOP | |
| 32 END | Конец |

Программа записи в ПЗУ 256 тетрад из любой области ОЗУ микроЭВМ с начальным адресом, находящимся в регистре R3, управляет и работой программатора в режиме записи (рис. 2).

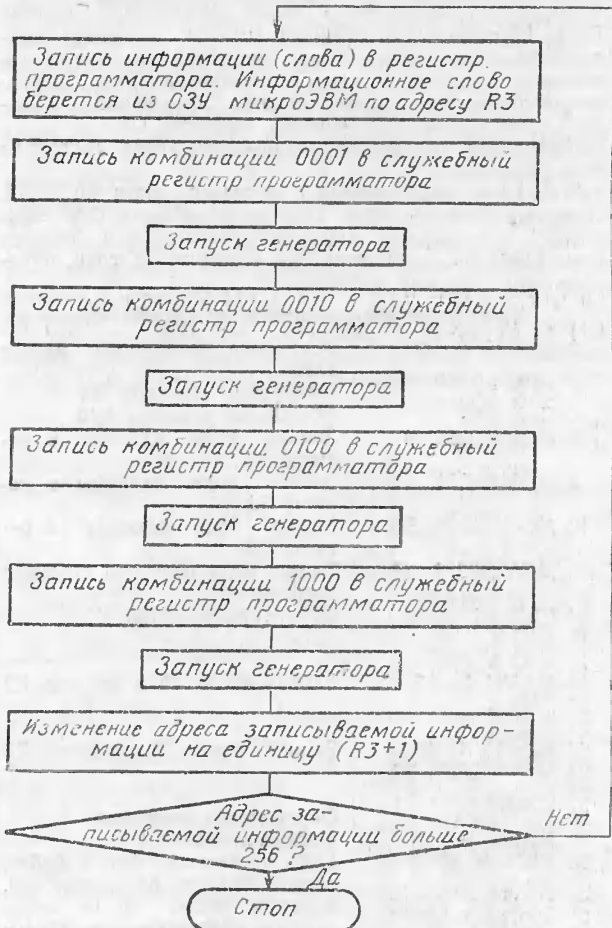


Рис. 2. Упрощенный алгоритм записи информации в ПЗУ

Программа записи информации в ПЗУ

| | |
|-----------------|--|
| 0 DIG 2 | |
| 1 DIG 5 | |
| 2 DIG 6 | |
| 3 MOVH X, R4 | Запись числа 256 в регистр R4 |
| 5 CLR R1 | Обнуление регистра R1 |
| 7 CLR R11 | Обнуление регистра R11 |
| 9 MARK 8 | Установка метки 8 |
| 11 MOV#0400, S2 | Запись 4-разрядного слова в информационный регистр программатора |
| 13 MOV @R3, S2 | |
| 15 OUTOWS | |
| 17 MOV#0700, S2 | Запись комбинации 0001 в служебный регистр программатора |
| 19 MOV#0800, S3 | |
| 21 OUTOWS | |
| 23 OUTO 0500 | Запуск генератора |

25 MOV#0400, S3 Запись комбинации 0010 в служебный регистр программатора

27 OUTOWS

29 OUTO 0500 Запуск генератора

31 MOV#0200, S3 Запуск комбинации 0100 в служебный регистр программатора

33 OUTOWS

35 OUTO 0500 Запуск генератора

37 MOV#0100, S3 Запись комбинации 1000 в служебный регистр программатора

39 OUTOWS

41 OUTO 0500 Запуск генератора

43 ADD#1, R1 Увеличение на единицу содержимого регистра R1

45 ADD#1, R3 Увеличение на единицу содержимого регистра R3

47 ADD#1, R11 Увеличение на единицу содержимого регистра R11

49 BSA R1, R4 Если содержимое регистров R1,

R4 одинаково, то конец программы. Если нет, то переход к метке 8.

51 JMM 8
53 STOP
55 END

Конец

Адрес для справок: 680028, г. Хабаровск, ул. Истомина, д. 88, кв. 53.

ЛИТЕРАТУРА

1. Кофрон Дж. Технические средства микропроцессорных систем. — М.: Мир, 1983.
2. Титце У., Шенк К. Полупроводниковая схемотехника. — М.: Мир, 1982.
3. Алексенко А. Г., Галицын А. А., Иванов А. Д. Проектирование радиоаппаратуры на микропроцессорах. — М.: Радио и связь, 1984.
4. Григорьев В. Л. Программное обеспечение микропроцессорных систем. — М.: Энергоиздат, 1983.
5. Техническое описание микроЭВМ «Электроника ДЗ-28».
6. Справочник программиста для микроЭВМ «Электроника ДЗ-28».

Статья поступила 15 мая 1986 г.

Исходное состояние: А — 8-разрядный множитель, ВС — 16-разрядный множитель. Конечный результат: А, HL — произведение, E — 00, B, C, D без изменений.

| | |
|------------|------------|
| LXI H,0000 | DAD B |
| MOV E, L | ADC E |
| ADD A | RET |
| ROT | ROT: MACRO |
| ROT | LOCAL Q |
| ROT | JNC Q |
| ROT | DAD B |
| ROT | ADC E |
| ROT | Q: DAD H |
| ROT | ADC A |
| RNC | END M |

Время выполнения — 314 тактов. Данная подпрограмма значительно отличается от описанной [3] (316 тактов) и в 1,5 раза быстрее — в [2] (475 тактов).

3. Подпрограмма умножения двух 12-разрядных чисел без знака.

Исходное состояние: С — старшие 8 разрядов первого сомножителя (в старшей тетраде D — младшие 4 разряда первого сомножителя, в младшей тетраде D — старшие 4 разряда второго сомножителя); E — младшие 8 разрядов второго сомножителя. Конечный результат: А, HL — произведение, B — 0, старшая тетрада D — 0, содержимое С младшей тетрады D и E не изменяется.

| | |
|----------|------------|
| MVI B,00 | ROT |
| MOV A,D | ROT |
| ANI F0 | ROT |
| MOV H A | ROT |
| MOV A,D | ROT |
| SUB H | RNC |
| MOV D,A | DAD D |
| MOV L, B | ADC B |
| MOV A,C | RET |
| DAD H | ROT: MACRO |
| ADC A | LOCAL Q |
| ROT | JNC Q |
| ROT | DAD D |
| ROT | ADC B |
| ROT | Q: DAD H |
| ROT | ADC A |
| ROT | END M |

УДК 681.3.06

А. Б. Глазов, С. А. Костарев, Е. В. Суханова

ЭФФЕКТИВНЫЕ ПРОГРАММЫ УМНОЖЕНИЯ ДЛЯ МИКРОПРОЦЕССОРА КР580ИК80А

При построении цифровых фильтров, реализации быстрого преобразования Фурье и других процессов обработки данных на микропроцессорных системах значительная часть времени обработки расходуется на выполнение умножения чисел. Ниже приводится несколько подпрограмм умножения чисел без знака различной разрядности для микропроцессора КР580ИК80А, имеющих малое время выполнения. «Время выполнения N тактов» означает, что программа выполняется не более чем за N тактов.

Основная идея приведенных алгоритмов заключается в одновременности сдвига результата произведения и одного из множителей. Подобный сдвиг в каждой подпрограмме оформлен в виде макрокоманды ROT и описан после соответствующей подпрограммы.

Ввиду краткости подпрограмм и их регулярности комментарии к ним исчерпываются замечаниями, сделанными выше. Для начинающих программистов в дальнейшем комментарии сопровождают только макрокоманды. Из этих же соображений описание макрокоманд приводится после текста подпрограмм.

1. Подпрограмма умножения двух 8-разрядных чисел без знака.

Исходное положение: первый сомножитель находится в регистре H, второй в — E. Конечный результат: HL — произведение, D — 0, A, B, C, E не изменяются.

| | |
|----------|-------|
| MVI D,00 | ROT |
| MOV L,D | ROT |
| DAD H | RNC |
| ROT | DAD D |
| | RET |

| | |
|------------|---|
| ROT: MACRO | ; описание макрокоманды |
| LOCAL Q | ; метка Q своя для каждой макрокоманды |
| JNC Q | ; тело макрокоманды; если анализируемый бит первого сомножителя равен 1, |
| DAD D | ; то к результату, накапливаемому произведению, прибавляется второй сомножитель |
| Q: DAD H | ; тель. Кроме того, осуществляется сдвиг результата и первого сомножителя на 1 разряд |
| END M | ; признак конца макрокоманды |

Замечание для ручного ассемблирования: в подпрограмму вместо каждого «ROT» вставляется только тело макрокоманды.

Время выполнения 257 тактов в 2 раза меньше времени выполнения одной из подпрограмм умножения, рассмотренной в [2] (520 тактов), и на 5% меньше времени выполнения «быстрого» умножения с помощью таблиц произведений тетрад, описанного в [1, 2] (270 и 273 такта) и требующего громоздкой таблицы произведений, занимающей 256 байт в ПЗУ.

Наиболее удобной для системы команд микропроцессора КР580ИК80А является:

2. Подпрограмма умножения 8-разрядного числа без знака на 16-разрядное.

Время выполнения — 509 тактов. Для сравнения заметим, что подпрограмма умножения двух 10-разрядных чисел, описанная в [1], имеет время выполнения, превышающее 1300 тактов, т. е. достигается выигрыш более чем в 2 раза.

4. Подпрограмма умножения двух 16-разрядных чисел без знака.

```

ROT2      LXI H,00
ROT2      MOV A,D
JNC G     MOV D,L
INR D     ADD A
G: MOV E,A
POP PSW
RNC
DAD B
RNC
INX D
RET
ROT1: MACRO
LOCAL P
JNC P
DAD B
ADC D
P: DAD H
ADC A
END M
ROT2: MACRO
LOCAL N
JNC N
DAD B
ADC E
N: DAD H
ADC A
END M

```

Исходное состояние: BC — первый сомножитель, DE — второй сомножитель. Конечный результат: DE, HL — произведение, BC не изменяется.

Время выполнения — 695 тактов. Это на 12 % лучше, чем у подпрограммы, описанной в [3] (781 такт), и почти в 3 раза быстрее подпрограммы умножения 16-разрядных чисел [4] (2060 тактов). Текст этой подпрограммы в кодах, размещенный начиная с адреса 0800, имеет вид

```

0800 21 00 00 7A 55 87 D2 0B 08 09 8A 29 8F D2 12 08
0810 09 8A 29 8F D2 19 08 09 8A 29 8F D2 20 08 09 8A
0820 29 8F D2 27 08 09 8A 29 8F D2 2E 08 09 8A 29 8F
0830 D2 35 08 09 8A 29 8F D2 3C 08 09 8A 57 7B 1E 00
0840 1F F5 87 29 8F D2 4A 08 09 8B 29 8F D2 51 08 09
0850 8B 29 8F D2 58 08 09 8B 29 8F D2 5F 08 09 8B 29
0860 8F D2 66 08 09 8B 29 8F D2 6D 08 09 8B 29 8F D2
0870 74 08 09 8B 29 8F D2 7A 08 14 5F F1 D0 09 D0 13
0880 C9

```

Как видно из изложенного, одновременный сдвиг сомножителя и результата в системе команд микропроцессора КР5801К80А позволяет писать эффективные подпрограммы умножения, которые при сопоставимой длине подпрограммы существенно (примерно в 2—3 раза) быстрее широкоизвестных [1, 2, 4] и значительно экономят память (более 200 байт).
Телефон для справок: 3-63 31 (г. Сыктывкар).

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. — М.: Радио и связь, 1984. — 272 с.
2. Григорьев В. Л. Программное обеспечение микропроцессорных систем. М.: Энергоатомиздат, 1983. — 208 с.
3. Бернар Купе. Сверхэффективные программы умножения для

микропроцессоров i 8080 и Z80. — Электроника, 1983, т. 56, № 6, с. 74—76.

4. Цифровые фильтры и устройства обработки сигналов на интегральных микросхемах / Под ред. Б. Ф. Высоцкого. — М.: Радио и связь, 1984. — 216 с.

Статья поступила 31 мая 1985 г.

СРЕДСТВА ОТЛАДКИ

УДК 681.385:621.317.799

Е. Д. Баран, Е. И. Кошелева, Е. М. Салмина

АНАЛИЗАТОР МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Комплексная отладка аппаратной основы и программного обеспечения (ПО) микропроцессорных систем (МПС) — один из заключительных этапов разработки и производства аппаратуры. На этом этапе при выполнении программ в реальном времени обнаруживаются неисправности, обусловленные чувствительностью БИС к тактовой частоте, последовательностям кодов, конструктивным особенностям устройств, проявляются ошибки, не замеченные при раздельном проектировании программ и технических средств [1—3].

Обнаружить и идентифицировать подобные дефекты МПС можно только при использовании специализированных контрольно-диагностических устройств — логических анализаторов (АЛ) [4—10].

Современные универсальные АЛ предоставляют оператору возможность не только анализировать последовательности состояний контролируемого объекта, но и исследовать временные соотношения сигналов, определять некоторые количественные характеристики прикладных программ, отображать результаты испытаний в форме, наиболее удобной для их интерпретации в соответствии с целями диагностического эксперимента. Таким АЛ свойственны модульный принцип построения аппаратуры, развитое программное обеспечение, возможность адаптации АЛ к испытаниям МПС различных структур и характеристик, комплекс средств,

облегчающих работу оператору по подготовке и проведению испытаний [11—15].

Рассматриваемый анализатор АЛ-6 относится к классу приборов [16, 17], предназначенных для анализа логических состояний. На основе АЛ-6 можно реализовать различные системы, отличающиеся набором выполняемых функций и техническими характеристиками. С «зашитым» ПО АЛ-6 предназначен для анализа информационных потоков в МПС с интерфейсом типа «Общая шина» и протоколом обмена микроЭВМ «Электроника С5-21М» [18].

Состав и технические характеристики АЛ-6

Преобразуют, отображают и документируют данные, а также управляют АЛ микропроцессорные функциональные модули (МФМ) семейства «Электроника С5-21М» и стандартные периферийные устройства. Для требуемого быстрого действия селекция и запоминание информации реализуются специально разработанными аппаратными средствами — модулем логической обработки (МЛО) и модулем оперативного ЗУ (ОЗУА).

Структурная схема базовой модели АЛ (рис. 1) состоит из минимально необходимого комплекта модулей:

— двух микроЭВМ «Электроника С5-21М», выполняющих функции контроллера пульта (КП) и контроллера анализатора (КА);

— МФМ «Электроника С5-2113» — ЗУ программ АЛ (модуль содержит ПЗУ масочного типа (8К слов), ЭППЗУ с ультрафиолетовым стиранием — 4 (8)К слов и программатор ЭППЗУ);

— МФМ «Электроника С5-2106» — дисплейного адаптера (ДА) черно-белого или цветного видеоконтрольного устройства (ВКУ); модуль обеспечивает отображение в кадре (2048 символов) кода ASCII од-

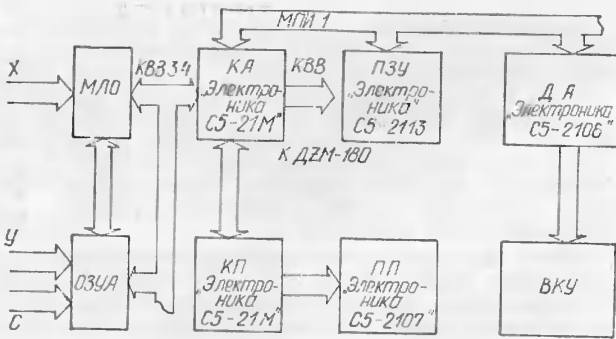


Рис. 1. Структурная схема анализатора АЛ-6

ной из восьми страниц текстовой информации или одной из двух страниц графической информации (512×256 точек). Емкость внутреннего ОЗУ адаптера — 16К слов;

— МФМ «Электроника С5-2107» — пульта программиста (ПП), предназначенного для подготовки и отладки программы в машинных кодах, в анализаторе, выполняющего функции пульта управления;

— ВКУ «Квант»;
— алфавитно-цифрового печатающего устройства ДЗМ-180.

Контроллер АЛ соединяется со стандартными МФМ (2106 и 2113) через межмодульный параллельный унифицированный интерфейс МПИ1, а с контроллером пульта — последовательным каналом. Модули МЛО и ОЗУА соединяются с контроллером АЛ через два канала ввода-вывода (КВВ) микроЭВМ. При таком соединении адреса ячеек памяти и служебных регистров этих модулей не занимают места в адресном поле КА. Это особенно важно при увеличении количества модулей в системе. Кроме того, при соединении через каналы ввода-вывода можно заменять одноплатами микроЭВМ на однокристалльные практически без изменения схемных решений и программ обслуживания.

Группы входов анализатора предназначены для подключения сигналов, формируемых контролируемой системой: группа входов «С» — для сигналов синхронизации, «Х» — для данных, содержащих контролируемые признаки, «У» — для регистрируемых данных. В зависимости от структуры испытуемого объекта и задачи испытаний группы входов можно объединить. В анализаторе АЛ-6, предназначенном для отладки систем с интерфейсом «Общая шина», объединены входы Х и У.

В базовом исполнении комплект модулей анализатора и источник питания БПСБ — 1 устанавливаются



Рис. 2. Внешний вид макета анализатора в составе системы отладки программ микроЭВМ «Электроника С5-21М»

ся в корпус типа «Электроника С МС91301» размерами 300×298×165 мм (рис. 2).

Основные технические характеристики базовой модели анализатора АЛ-6

| | |
|--|--|
| Количество входов: | 28 |
| признаков запуска | 28 |
| записи данных | 8 |
| синхронизации | 4 |
| Количество одновременно отслеживаемых признаков | 4 |
| Маскирование: | |
| входов признаков | Поразрядное, общее для всех признаков, с возможностью задания маски только на время записи |
| входов синхронизации | Поразрядное, для четырех входов |
| Емкость памяти азинных (ОЗУА), 28 разрядных слов | 256 |
| Минимальная длительность цикла записи, нс | 200 |
| Задержка запуска, тактов/признаков | С...10 ⁶ |
| Модули МЛО и ОЗУА выполнены на элементах ТТЛ и ТТЛШ-типов. | |

Принцип действия

Основная функция АЛ-6 — селективный отбор интересующей оператора информации — выполняется по заданным предварительно совокупностям интерфейсных, функциональных и алгоритмических признаков.

Для пояснения принципа действия воспользуемся упрощенной функциональной схемой анализатора (рис. 3) и временными диаграммами (рис. 4, 5). На схеме показаны только блоки и связи модулей МЛО

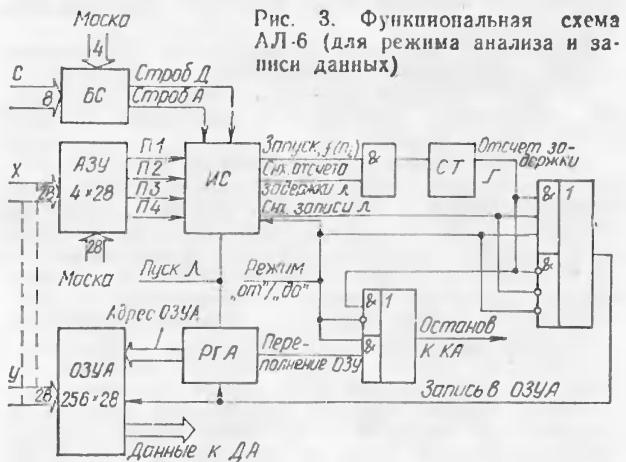


Рис. 3. Функциональная схема АЛ-6 (для режима анализа и записи данных)

и ОЗУА, участвующие в сборе данных. Обмен информацией в контролируемой системе осуществляется по совмещенной шине «адрес-данные» (АД) и реализуется с помощью набора интерфейсных сигналов (СИА, СИП и др., рис. 4). По этим сигналам блок синхронизации БС (см. рис. 3) формирует строб-импульсы для селекции контролируемых данных в соответствии с их

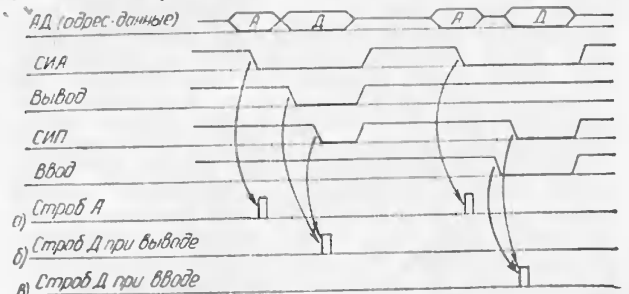


Рис. 4. Формирование сигналов блоком синхронизации АЛ-6

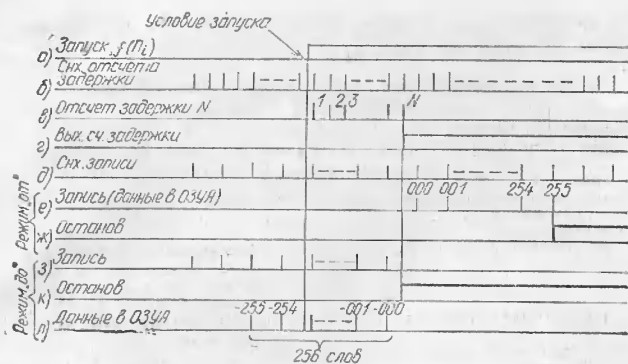


Рис. 5. Временные диаграммы АЛ-6

интерфейсными признаками — признаками адреса или данных, чтения или записи и т. п. Информация, появляющаяся на шине АД в процессе испытаний, поступает в анализатор через объединенные группы входов X и Y и отбирается с помощью блоков ассоциативного ЗУ (АЗУ) и индикатора совпадений (ИС) в соответствии с функциональными и алгоритмическими признаками, т. е. по принадлежности к определенной зоне адресов, по конкретному виду команд (сравнения, перехода, сложения и т. п.). При этом учитывается последовательность событий, например, появление определенных данных в определенных адресах, выполнение команды перехода после получения заданного результата и пр.

Признаками для этого этапа анализа информации являются записанные предварительно в АЗУ коды.

Последовательности событий распознаются блоком ИС по сигналам совпадения с признаками П1..П4, поступающими из АЗУ.

Результат анализа данных блоками БС, АЗУ, ИС используется для определения моментов начала или окончания записи в ОЗУ. По аналогии с запуском развертки осциллографа эти моменты обычно называют моментами запуска АЛ или просто запуском, но в отличие от осциллографа выделяют два принципиально различных режима в соответствии с тем, является ли момент выполнения условий запуска началом или окончанием записи данных в ОЗУ.

Условиями запуска можно выбрать: появление на шине АД определенного кода адреса или последовательности адресов, выборку определенной команды в определенном адресе и т. д. Иначе говоря, условие запуска — функция $f(P_i)$ от вводимых в АЗУ кодов признаков и режима работы ИС.

Если выбран режим записи данных, поступающих после выполнения условия запуска, то сигнал окончания записи — переполнение ОЗУ. Такой режим, называемый режимом «положительного» запуска или режимом записи «ОТ», позволяет контролировать последствия некоторого события, выбранного в качестве условия запуска (рис. 5, е, ж). В режиме «ДО» («отрицательный» запуск) запись данных начинается сразу после пуска контролируемой системы, в процессе записи ОЗУ может неоднократно переполняться до реализации условий запуска, после чего запись прекращается (рис. 5, з, к, л). Таким образом, можно контролировать предисторию некоторого события.

Для упрощения идентификации интересующих оператора участков контролируемых последовательностей начало записи (в режиме «ОТ») или окончание записи (в режиме «ДО») могут быть задержаны относительно момента запуска. В последнем случае запоминаются и могут быть проанализированы данные в окрестностях условия запуска. Задержка (N) задается количеством элементарных тактов обмена (сроб-импульсов БС) или выбираемых заранее событий (рис. 5, б, в, г) и отсчитывается счетчиком СТ (рис. 3).

В режиме селективной записи запись в ОЗУ производится только при появлении в контролируемой последовательности кодов признаков, хранимых в АЗУ (рис. 5, е, з). Сигналы синхронизации счета задержки и разрешения селективной записи формируются блоками БС, АЗУ и ИС. Информация, накопленная в ОЗУ, по сигналу «Останов» переписывается в ОЗУ дисплейного адаптера, анализатор переводится в режим отображения данных в выбранном формате.

Спецификация режимов

Более детально функциональные возможности и технические характеристики АЛ-6 описываются двумя спецификациями режимов анализатора, которые отображаются на экране ВКУ после включения и начального пуска анализатора (рис. 6, 7).

ЛИСТО

ПРОГРАММА ЗАПИСИ СОСТОЯНИЙ 'ОБЩЕЙ ШИНЫ' 'С5'

ЗАПИСЬ АДРЕСОВ и ДАННЫХ ПРИ ВВОДЕ M_0, M_1, M_2 , ВЫВОДЕ M_0, M_1, M_2

ОТ или ДО ПОСЛЕДОВАТЕЛЬНОСТИ ПРИЗНАКОВ:

1. X 2. П1 V П2 V П3 V П4 3. П1 & П2 & П3 & П4
 4. П1 (T) & П2 (T) & П3 (T+1) & П4 (T+1) 5. П1 (T) & П2 (T) & П3 (T+X) & П4 (T+X)

ПОСЛЕ ОТСЧЕТА ПРИЗНАКОВ: 1. X 2. П1 3. П2 4. П3 5. П4
 6. П1 V П2 V П3 V П4 7. НК 000001 PAZ

ПРИ ПОЯВЛЕНИИ ПРИЗНАКОВ: 1. X 2. П1 3. П3 4. П1 V П3 5. П2 V П4
 6. П1 7. П3 8. П1 V П3 9. П1 V НК

ГРУППЫ ВХОДОВ АЛ - 6 АД АД Е ЗП НК PPP
 НОМЕР РАЗРЯДА 0-----7 8-----15 0-----7 0 1 2
 ЛОГИКА, ФОРМАТ '2,8,10,16' \pm \pm \pm \pm

ЗНАЧЕНИЯ) АДРЕС П1:= 9996 00 0 0 0
 ПРИЗНАКОВ) ДАННЫЕ П2:= 8056 00 0 0 0
) АДРЕС П3:= 999A 00 0 0 0
) ДАННЫЕ П4:= 8020 00 0 0 0
 МАСКА ПРИ ЗАПУСКЕ, ЗАПИСИ 11110000 00000000 11111111 0 1 1

Рис. 6. Лист 0, формируемый после начального пуска анализатора *

Первая спецификация (лист 0, см. рис. 6) рассчитана на инженеров, проводящих комплексные испытания МПС. В оформлении ее учтены особенно-

ЛИСТ 4

КОНТРОЛЬ СОСТОЯНИЯ ШИН АД (АДРЕС-ДАННЫЕ)

ФОРМАТ '8,16'

A = (СЧК), АП

A (A) A (A)
 9996 8056 999A 8020

ЗАПУСК ОТ, ДО A, (A) \square \square \square \square

ОТСЧЕТ 000001 PAZ --, -- * * * \square

ЗАПИСЬ:
 1. A, (A) ПРИ A, JA * *
 2. A, (A) ПРИ (A) * *
 3. (СЧК), (СЧК) и A, (A) ПРИ A \square
 4. (OP) В ЗАДАЧИ

МАСКА A, (A) ПРИ ЗАПУСКЕ, ЗАПИСИ 11110000 00000000

Рис. 7. Лист 4. Спецификация программиста

* Здесь и далее фрагменты кадра, окаймленные рамкой, на экране ВКУ отображаются в инверсном под-свете

сти интерфейса МПС на основе микроЭВМ и МФМ семейства «Электроника С5 21М». Язык спецификаций прост, использует общепринятые термины, обозначения интерфейсных сигналов и операторов.

На рис. 6 М0, М1, М2 — магистрали интерфейсов микроЭВМ — соответственно внутреннего и системных — МПИ1, МПИ2; X — произвольное событие или число; НК — синхронимпульс «Начало команды»; П — инверсия признака П; ПРР — сигнал «Разрешение прерывания».

На листе 0 разрешенные для выбора или коррекции разновидности режимов запуска, счета задержки, записи, формат и логика отображения данных, а также значения задержки, признаков и кода маски признаков отмечены в спецификации подчеркиванием соответствующих знаменосцев. Выбранные разновидности и параметры режимов (программа работы анализатора) выделяются инверсным подсветом. Программа самоконтроля анализатора (см. рис. 6) формируется автоматически после начального пуска и представляет собой контрольную задачу для комплексной проверки работы блоков управления и памяти анализатора.

Этой программой предписывается запись адресов и данных, появляющихся в циклах ввода и вывода на всех магистралях микроЭВМ (М0, М1, М2). Запись в ОЗУ АД должна начаться после того, как в одном цикле обмена в некоторый момент времени Т появится адрес П1 с данными П2, а через произвольное число тактов Х появится следующая пара признаков — данные П4 в адресе П3. Задержка начала записи относительно указанного условия запуска равна интервалу времени до появления одного признака П4, а сигнал разрешения записи формируется при появлении признака П1 или сигнала НК (селективная запись по событию П1 V НК). Значения отслеживаемых признаков П1..П4 и код маски признаков указаны в нижних строках кадра. Маска задается в двоичном коде. Формат отображения записываемых данных — шестнадцатеричный, полярность логики — отрицательная.

По выполнении программы результаты записи в краткая характеристика выбранного режима работы отображаются на первом листе дисплейного ОЗУ (рис. 8). Возможен выбор адреса ОЗУА, с которого начинается чтение и отображение записанных данных.

ЛИСТ 1

СОСТОЯНИЕ МАГИСТРАЛИ

ОТ ПОСЛЕДОВАТЕЛЬНОСТИ П1(Т)XП2(Т)XП3(Т+X)XП4(Т+X)

| ОТСЧЕТА | ОСОСО1 | РАЗ | ПЧ | ПРИ ПОЯВЛЕНИИ | | | | П1VНК |
|---------------------------|-------------|----------|----------|---------------|----|----|--|-------|
| ГРУППЫ ВХОДОВ АД - 6 | АД | АД | Е | ЗП | НК | ПР | | |
| НОМЕР РАЗРЯДА | 0-----7 | 8-----15 | 0-----7 | 0 | 1 | 2 | | |
| ЛОГИКА, ФОРМАТ | '2,8,10,16' | -, '16' | -, '16' | -, '2' | | | | |
| АДРЕС П1:= | | 9996 | 00 | 0 | 0 | 0 | | |
| ДААННЫЕ П2:= | | 8056 | 00 | 0 | 0 | 0 | | |
| АДРЕС П3:= | | 999A | 00 | 0 | 0 | 0 | | |
| ДААННЫЕ П4:= | | 8020 | 00 | 0 | 0 | 0 | | |
| МАСКА ПРИ ЗАПУСКЕ, ЗАПИСИ | 11110000 | 00000000 | 11111111 | 1 | 1 | 1 | | |

| АД | Е | ЗП | НК | ПРР | АД | Е | ЗП | НК | ПРР |
|----------------------|---|----|----|-----|----------------------|---|----|----|-----|
| +000 A 22FE 00 0 1 0 | | | | | +010 A 230C 00 0 1 0 | | | | |
| +001 Д 83BF 00 0 1 0 | | | | | +011 Д 1180 00 0 1 0 | | | | |
| +002 A 3996 00 0 0 0 | | | | | +012 A 2310 00 0 1 0 | | | | |
| +003 Д 8056 00 0 0 0 | | | | | +013 Д 308E 00 0 1 0 | | | | |
| +004 A 2302 00 0 1 0 | | | | | +014 A 2312 00 0 1 0 | | | | |
| +005 Д 51B0 00 0 1 0 | | | | | +015 Д F7E8 00 0 1 0 | | | | |
| +006 A 2306 00 0 1 0 | | | | | +016 A 22FC 00 0 1 0 | | | | |
| +007 Д E702 00 0 1 0 | | | | | +017 Д 84AF 00 0 1 0 | | | | |
| +008 A 230A 00 0 1 0 | | | | | +018 A 22FE 00 0 1 0 | | | | |
| +009 Д 91AD 00 0 1 0 | | | | | +019 Д 835F 00 0 1 0 | | | | |

Рис. 8, а. Лист 1. Краткая характеристика выбранного режима работы анализатора. Содержимое ячеек ОЗУА в шестнадцатеричном формате

ЛИСТ 2

СОСТОЯНИЕ МАГИСТРАЛИ

| АД | Е | ЗП | НК | ПРР | АД | Е | ЗП | НК | ПРР |
|------------------------|---|----|----|-----|------------------------|---|----|----|-----|
| +020 A 034626 00 0 0 0 | | | | | +040 A 021402 00 0 1 0 | | | | |
| +021 Д 100126 00 0 0 0 | | | | | +041 Д 050560 00 0 1 0 | | | | |
| +022 A 021402 00 0 1 0 | | | | | +042 A 021406 00 0 1 0 | | | | |
| +023 Д 050560 00 0 1 0 | | | | | +043 Д 163402 00 0 1 0 | | | | |
| +024 A 021406 00 0 1 0 | | | | | +044 A 021412 00 0 1 0 | | | | |
| +025 Д 163402 00 0 1 0 | | | | | +045 Д 110655 00 0 1 0 | | | | |
| +026 A 021412 00 0 1 0 | | | | | +046 A 021414 00 0 1 0 | | | | |
| +027 Д 110655 00 0 1 0 | | | | | +047 Д 010600 00 0 1 0 | | | | |
| +028 A 021414 00 0 1 0 | | | | | +048 A 021420 00 0 1 0 | | | | |
| +029 Д 010600 00 0 1 0 | | | | | +049 Д 030216 00 0 1 0 | | | | |
| +030 A 021420 00 0 1 0 | | | | | +050 A 021422 00 0 1 0 | | | | |
| +031 Д 030216 00 0 1 0 | | | | | +051 Д 173750 00 0 1 0 | | | | |
| +032 A 021422 00 0 1 0 | | | | | +052 A 021374 00 0 1 0 | | | | |
| +033 Д 173750 00 0 1 0 | | | | | +053 Д 102257 00 0 1 0 | | | | |
| +034 A 021374 00 0 1 0 | | | | | +054 A 021376 00 0 1 0 | | | | |
| +035 Д 102257 00 0 1 0 | | | | | +055 Д 101677 00 0 1 0 | | | | |
| +036 A 021376 00 0 1 0 | | | | | +056 A 034626 00 0 0 0 | | | | |
| +037 Д 101677 00 0 1 0 | | | | | +057 Д 100126 00 0 0 0 | | | | |
| +038 A 034626 00 0 0 0 | | | | | +058 A 021402 00 0 1 0 | | | | |
| +039 Д 100126 00 0 0 0 | | | | | +059 Д 050660 00 0 0 1 | | | | |

Рис. 8, б. Лист 2. Содержимое ячеек ОЗУА в восьмеричном формате

В рассматриваемом примере группа входов Е не используется.

Вторая спецификация (лист 4, см. рис. 7) рассчитана на инженеров-программистов. Она оформлена в виде таблицы, в которой использованы обозначения: А — адрес — содержимое счетчика команд (СЧК) или адрес памяти (АП); ОР — общие регистры. Как и на листе 0, разрешенные для выбора разновидности режимов и их параметры подчеркнуты, а выбранные компоненты программы анализа выделены инверсным подсветом. Отслеживаемая последовательность появления признаков запуска выделяется инверсным подсветом знака в группе разновидностей режимов запуска.

Пример — программа (см. рис. 7) самоконтроля анализатора, режимы и параметры которой совпадают с программой на рис. 6. Результаты сбора информации

ЛИСТ 5

| (СЧК) | [(СЧК)] | АП | (АП) |
|-------------------|---------|------|------|
| +000 ЧТ 22FE 83BF | | | |
| +002 ЧТ | | 3996 | 8056 |
| +004 ЧТ 2302 51B0 | | | |
| +006 ЧТ 2306 E702 | | | |
| +008 ЧТ 230A 91AD | | | |
| +010 ЧТ 230C 1180 | | | |
| +012 ЧТ 2310 308E | | | |
| +014 ЧТ 2312 F7E8 | | | |
| +016 ЧТ 22FC 84AF | | | |
| +018 ЧТ 22FE 83BF | | | |
| +020 ЧТ | | 3996 | 8056 |
| +022 ЧТ 2302 51B0 | | | |
| +024 ЧТ 2306 E702 | | | |
| +026 ЧТ 230A 91AD | | | |
| +028 ЧТ 230C 1180 | | | |
| +030 ЧТ 2310 308E | | | |
| +032 ЧТ 2312 F7E8 | | | |
| +034 ЧТ 22FC 84AF | | | |
| +036 ЧТ 22FE 83BF | | | |
| +038 ЧТ | | 3996 | 8056 |

Рис. 9. Лист 5. Отображение данных при работе со спецификацией программиста

пии о ходе программы отображаются на листе 5 дисплейного ОЗУ в форме, более привычной для разработчика программ (рис. 9) или в формате, используемом на листе 1 (см. рис. 8).

Управление анализатором

Работой анализатора управляют с пульта оператора (рис. 10). Программа работы АЛ подготавливается в экранном режиме на одной из спецификаций с помощью клавиш перемещения маркера (↓, ↑, →, ←), выбора режима (□ ■) и записи цифровой информации (0, 1, ..., F, ЗП).

| | | | | | | | |
|---|---|---|---|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | → | ← | АВТ | ЧТА |
| 4 | 5 | 6 | 7 | ↓ | ↑ | ЗП | №Л |
| 8 | 9 | А | В | ПСК | ■□ | МА | ТСТ |
| С | Д | Е | F | ↑ИН | ↓ИН | ОД | ПЧТ |

Рис. 10. Клавиатура анализатора АЛ-6

В режим сбора, отображения данных или самконтроля, в состояние останова анализатор переводится клавишами «Пуск», «Чтение ОЗУА», «Тест», «Останов динамический» (обозначения ПСК, ЧТА, ТСТ, ОД).

Для смены формы спецификации или массива отображаемых данных служат клавиши смены номера экранного листа (0, 1, ..., 7, № Л) и клавиши «перелистывания» ОЗУА (↑ИН, ↓ИН).

Клавиша МА предназначена для отображения информации в графической форме — в виде матрицы адресов. В отображаемой на экране квадратной матрице координаты каждой точки соответствуют определенному (записанному в ОЗУА) коду адреса. Таким образом, в одном кадре можно оценить правильность прохода сравнительно большого фрагмента отслеживаемой программы по адресному полю (см. матрицу адресов программы самоконтроля на рис. 11).

Информация на DZM-180 (клавиша «Печать» — ПЧТ) выводится в виде «копии» с экранного листа, в виде таблицы по восемь слов в строке с печатью «круглого адреса» или в форме столбца «Адрес-данные» (формат печати, объем и адрес начала регистрируемого массива данных выбирается макрокодом).

Отметим, что анализатором управляют в режиме интерпретации клавиш пульта. Это сохраняет отладочные функции пульта «Электроника С5-2107» и делает возможным построение на базе АЛ-6 системы отладки программ микроЭВМ «Электроника С5-21М». Для этого предусмотрен автономный режим работы

МАТРИЦА АДРЕСОВ

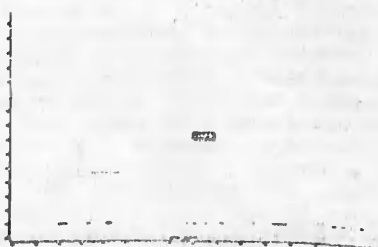


Рис. 11. Матрица адресов

модулей МЛО и ОЗУА (клавиша АВТ), при реализации которого входы АЛ соединяются с магистралью МПИ1 (или МПИ2) контроллера анализатора, выполняющего отлаживаемую программу, а предварительно подготовленные модули МЛО и ОЗУА, как обычно, «следят» за отладкой. При необходимости к КА можно подключить дополнительные модули ОЗУ, адаптеры ввода-вывода и др.

Автономность МЛО и ОЗУА при сборе данных позволяет встраивать эти модули в любую систему на основе микроЭВМ «Электроника С5-21М» (например, в комплекс 15УТ-1) и при наличии соответствующих программных средств использовать для контроля в реальном времени функционирования систем при отладке и эксплуатации.

Программное обеспечение

Программное обеспечение (ПО) анализатора аппаратно-ориентировано и обеспечивает работу системы «Оператор—пульт—модули—МЛО, ОЗУА—дисплей, печать». ПО АЛ-6 состоит из стандартных резидентных программ микроЭВМ «Электроника С5-21М» и прикладных программ. Резидентная часть ПО (3К слов) — программы расширения систем команд «резидентная тысяча» и программы обслуживания пульта размещены в БИС ПЗУ контроллеров — КП и КА. Прикладное ПО (7К слов) хранится в ПЗУ МФМ «Электроника С5-2113» и содержит программу начального пуска и пакет программ управления.

Программы управления запускаются по прерыванию от клавиш пульта или по командам передачи управления и обеспечивают:

- интерпретацию клавиш пульта;
- перемещение маркера и редактирование информации на экране дисплея;
- загрузку служебных регистров и контроль регистров состояний модулей МЛО и ОЗУА;
- преобразование информации для отображения и документирования в заданном формате;
- тестирование модулей МЛО и ОЗУА;
- формирование «подказок» оператору о порядке действий, сообщений об ошибках оператора, о текущем состоянии анализатора и о появлении в контролируемой программе отслеживаемой последовательности признаков.

Применение анализатора

Область применения АЛ-6 не ограничивается отладкой систем с интерфейсом «Общая шина». Для подготовки анализатора к испытаниям систем с другим интерфейсом дорабатывается блок синхронизации (или программ управления им) и изменяются обозначения входов и сигналов в спецификациях и листах отображения данных. Для увеличения количества входов признаков и записи, увеличения емкости памяти, усложнения режимов запуска и записи данных в АЛ-6 можно ввести дополнительные модули МЛО и ОЗУА.

В аппаратных средствах АЛ-6 предусмотрена возможность количественно оценивать некоторые характеристики контролируемых программ — подсчет числа событий, измерение временных интервалов между событиями и т. п.

Модульный принцип построения, возможность агрегатирования модулей МЛО и ОЗУА, разнообразие режимов селективного сбора и форм отображения данных и программируемый блок синхронизации позволяют гибко модифицировать функциональные возможности анализатора. Отладка значительной части ПО АЛ-6, комплексная отладка и испытания анализатора выполнены с его же помощью. Анализатор может служить основой для создания автоматизированной системы контроля и диагностирования микропроцессорных систем в условиях производства.

Адрес для справок: 630087, Новосибирск, пр. К. Маркса, 20. Новосибирский электротехнический институт. Телефон — 46-08-53.

ЛИТЕРАТУРА

1. Барраклаф, Цзян, Сол. Методы тестирования микромашинных устройств // ТИИЭР.— 1976.— Т. 64.— № 6.— С. 134—143.
2. Хавкин В. Е., Барашенков Б. В., Вершинин С. О. Контроль и диагностика микроЭВМ.— Обзоры по электронной технике. Сер. 3: Микроэлектроника. М.: ЦНИИ «Электроника», 1979.— 82 с.
3. Гобземис А. Ю., Удалов В. И. Методы тестового контроля микропроцессорных устройств // Автоматика и вычислительная техника.— 1978.— № 6.— С. 18.
4. Сантонн А. Совершенствование тестеров для наладки микропроцессорных систем // Электроника.— 1976.— Т. 49.— С. 26—29.
5. Фарли Б. Логические анализаторы или системы проектирования? // Электроника.— 1979.— Т. 52.— № 19.— С. 62—70.
6. Маршалл М. Возможности новейших анализаторов временных последовательностей // Электроника.— 1979.— Т. 52.— № 7.— С. 43—51.
7. Даматов А. М., Никитюк Н. М., Шюсслер Р. Применение микропроцессоров в приборостроении и в экспериментальных исследованиях (обзор) // ПТЭ.— 1979.— № 4.— С. 7—26.
8. Шлимович Е. М. Логические анализаторы для проверки и наладки сложных цифровых устройств и систем // Вопросы радиоэлектроники. Сер. ЭВТ.— 1980.— Вып. 4.— С. 27—42.
9. Анализатор логический 16 канальный 806 // Проспект ВДНХ СССР: ЦООНТИ «ЭКОС», 1978.
10. Анализатор логический тридцатидвухканальный 821.— Проспект ВДНХ СССР: ЦООНТИ «ЭКОС», 1984.
11. Коэч. 18-канальный логический анализатор стоимостью 4000 долларов // Электроника.— 1982.— Т. 55.— № 3.— С. 21—22.
12. Палквист С., Херен Г. Логический анализатор, генерирующий испытательные сигналы // Электроника.— 1981.— Т. 54.— № 18.— С. 26—35.
13. Универсальный логический анализатор // Электроника.— 1982.— Т. 55.— № 1.— С. 133—134.
14. Горовой В. Р., Бокарев А. В. и др. Комплекс диагностирования «Электроника НЦ-603» // 5-е Всесоюзное совещание «Техническая диагностика»: Тез. докл./Институт проблем управления.— М.: ИПУ, 1982.— С. 62—64.
15. Анализатор потока цифровых данных 823.— Проспект ВДНХ СССР. М.: ЦООНТИ «ЭКОС», 1984.
16. Баран Е. Д., Мирошников Е. Н., Чайка А. А. Многофункциональный логический анализатор // ПСУ.— 1982.— № 9.— С. 34—35.
17. А. с. 1032457 (СССР). Логический анализатор/Е. Д. Баран // Открытия. Изобретения.— 1983.— № 28.
18. Гальперин М. П., Кузнецов Ю. Я., Маслеников Ю. А. и др. МикроЭВМ «Электроника С5» и их применение. М.: Сов. радио, 1980.— 160 с.

Статья поступила 1 декабря 1985 г.

УДК 681.326.35.7

В. С. Безобразов, Б. И. Ларин, В. Ю. Сохранов, А. А. Шишкевич

ОТЛАДОЧНЫЙ КОМПЛЕКС ДЛЯ 32-РАЗРЯДНОЙ МИКРОЭВМ НА ОСНОВЕ ДВК-2

Для отладки аппаратуры микропроцессорных систем используются многоразрядные генераторы слов, логические и сигнатурные анализаторы, комплексы диагностирования [1—3]. Выпускаемые в настоящее время промышленностью логические анализаторы типа 806, 821, 832 [1], генераторы слов Г5-80 не могут в полном объеме удовлетворить разработчика аппаратных средств, особенно при использовании ИМС ТТЛ типа в устройствах с разрядностью больше 16 из-за ограниченных возможностей по числу каналов ввода-вывода (до 32) и быстродействию (частота дискретизации до 20 МГц).

Разработанные аппаратно-программные средства предназначены для отладки 32-разрядной микроЭВМ с быстродействием 1 млн. коротких операций в секунду, построенной на базе комплекта БИС КР1802.

Архитектура 32-разрядной микроЭВМ predeterminedла построение аппаратных средств отладки с большим числом (более 200) двунаправленных входо-выходов, работающих с ТТЛ-логическими уровнями. Отладочный комплекс обеспечивает возможность быстрого внесения корректив в процедуру отладки, а также отладку по этапам с переходом от

простых процедур к сложным, т. е. от отладки отдельных ячеек до отладки микроЭВМ в целом. Все ячейки микроЭВМ представляют собой законченные функциональные узлы (АЛУ, БМУ, ОЗУ и т. д.), связь между которыми осуществляется только по информационным магистралям и шинам признаков. Это позволяет отлаживать их отдельно, а затем собирать для окончательной отладки.

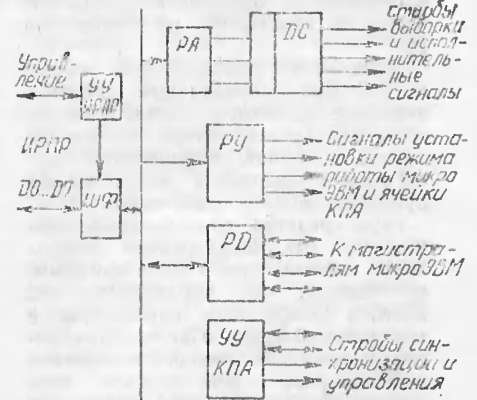
В состав комплекса входят: ДВК-2М; ячейка контроллера проверочной аппаратуры (КПА) — единственная ячейка, специально разработанная для отладки всех ячеек микроЭВМ, контактирующее устройство, в котором монтируется ячейка КПА и отлаживаемая ячейка. Конструкция контактирующего устройства позволяет осуществлять доступ к ячейкам как со стороны установочных элементов, так и со стороны печатного монтажа. При отладке микроЭВМ в целом ячейка КПА монтируется вместе с ячейками микроЭВМ в едином блоке. При этом увеличение длины магистралей и шин управления микроЭВМ минимально.

С помощью ДВК осуществляется оперативная подготовка программ и микропрограмм, контроль выполнения

программ в микроЭВМ, анализ ответных сигналов от микроЭВМ и вывод на экран дисплея результатов анализа.

Ячейка КПА предназначена для согласования разрядности и протоколов обмена между интерфейсом микроЭВМ и байтовым параллельным интерфейсом (ИРПР) ДВК-2М, выдачи входных воздействий на магистрали микроЭВМ, приема ответных кодов с магистралей микроЭВМ, задания режима работы отлаживаемой микроЭВМ (такт, автомат и др.).

Для реализации этих возможностей ячейка имеет в своем составе регистр управления и набор информационных регистров, реализованных на БИС КР1802ВВ1 и подключенных к информационным магистралям микроЭВМ (см. рисунок). При работе



Структурная схема ячейки контроллера проверочной аппаратуры

ячейка подключается к ИРПР ДВК вместо принтера переключателями на плате микроЭВМ «Электроника МС 1201.01» и задается вариант работы ИРПР со следующими адресами: регистр состояния источника — 177550; входной регистр — 177552; регистр состояния приемника — 177554; выходной регистр — 177556.

Обмен информацией с регистрами ячейки организован следующим образом: первое (или любое нечетное) обращение к ячейке КПА всегда является выводом байта информации из ДВК и воспринимается ячейкой как адрес информационного регистра или регистра управления. Этот адрес заносится в регистр адреса РА. В следующем (четном) обращении к ячейке КПА может осуществляться как запись информации в выбранный регистр, так и чтение из него. Несколько адресов в ячейке используется для формирования сигналов запуска, установки и других исполнительных сигналов, т. е. при обращении со стороны ДВК по такому адресу на выходе дешифратора формируется импульс длительностью 2...3 мкс.

При отладке микроЭВМ входные воздействия задаются загрузкой через ИРПР соответствующих кодов в необходимые для данной процедуры регистры ячейки КПА. После сигнала запуска с дешифратора происходит взаимодействие ячейки КПА и отлаживаемой ячейки: устройство управления КПА подключает информационные регистры к магистралям микроЭВМ, выдает входные коды в микроЭВМ с соответствующими сигналами синхронизации. Результат процедуры записывается обратно в регистры ячейки КПА. Далее через ИРПР происходит побайтное считывание результата в ДВК и его программный анализ. По результату анализа на экран дисплея выводится необходимая справочная информация.

Для организации работы с отлаживаемой микроЭВМ используется программное обеспечение, которое включает следующие составные части:

стандартную операционную систему ОС ДВК, содержащую монитор, экранные редакторы текста, трансляторы с макроассемблера и языков высокого уровня, компоновщик, системные библиотеки и ряд системных программ общего назначения;

кросс-средства, разработанные специально для 32-разрядной микроЭВМ, выполненные в виде программ, работающих под управлением ОС ДВК, и содержащие трансляторы с макроассемблера и с ассемблера, редактор связей, эмулятор. Результатом работы кросс-средств являются объектные или загрузочные файлы на внешнем носителе (НГМД), а также листинги их выполнения;

программы отладки функциональных узлов, отдельных ячеек и микроЭВМ в целом. Программы позволяют: вводить с внешнего носителя в микроЭВМ загрузочные файлы, созданные кросс-средствами; корректировать загрузочные модули и сохранять их в виде файла на внешнем носителе; при отладке отдельных ячеек задавать ряд тестов с возможностью вывода на дисплей сбойных ситуаций; обеспечивать доступ ко всем элементам хранения информации в микроЭВМ (регистры процессора, оперативная и микропрограммная память); при отладке микроЭВМ управлять режимами выполнения программ и микропрограмм.

Программное обеспечение разрабатывалось на ДВК-2М. Использование макроассемблера для создания программ отладки позволило получить

тесты, скорость выполнения которых на ДВК достаточна для устойчивой синхронизации осциллографа при наблюдении временных характеристик сигналов в отлаживаемой аппаратуре.

ЛИТЕРАТУРА

1. Васильев Н. П., Горовой В. Р. Микропроцессоры. Аппаратурно-программные средства отладки / Под ред. Л. Н. Преснухина // М.: Высшая школа, 1984.
2. Универсальный логический анализатор DAS9100 фирмы TEKTRONIX // Электроника, — 1982. — № 1. — С. 133—134.
3. Safferthal R. R. Adaptive Logik-analyse // Elektronik, — 1982. — N 22. — P. 77—82.

Справки по адресу: 103498, Москва, МИЭТ.

Статья поступила 19 мая 1986 г.

УДК 681.326

В. В. Тарасов

НЕСЛОЖНАЯ СИСТЕМА ОТЛАДКИ

Широкий контингент разработчиков использует микроЭВМ для реализации простых контроллеров. МикроЭВМ простых контроллеров характеризуется сравнительно малым объемом (до 1К байта) программного обеспечения (ПО) и выполняет несложные вычислительные задачи, функции обработки данных и управления. Это позволяет использовать для разработки подобной аппаратуры несложные в эксплуатации собственные инструменты проектирования и технику программирования в машинных кодах.

Простейший пример — это оценочный модуль, который позволяет только оценить возможности микропроцессорных средств, отладку же технических средств и комплексную отладку с его помощью провести нельзя [1].

Более сложные устройства, конструктивно выполненные в виде законченного блока с несколькими печатными платами, клавиатурой и индикатором, представляют пользователю гораздо больше возможностей. Так, учебно-отладочное устройство «Электроника-580» позволяет настраивать аппаратную часть целевого из-

делия и комплексно настраивать системы [2]. Тем не менее подобные средства имеют ряд недостатков.

Так, реализация пошагового режима с помощью прерывания затрудняет пользователю доступ к информации о состоянии микропроцессорных шин. Для такого доступа требуются дополнительные аппаратно-программные средства. Указанная информация необходима для специалистов по аппаратным средствам (схемотехников). Анализируя состояние шин, разработчик может делать выводы о работоспособности аппаратной части проектируемого изделия, а также тестировать в пошаговом режиме аппаратную часть статическими сигналами (метод тестирования статическими сигналами подробно изложен в работе [3]).

В большинстве отладочных устройств программы пользователя хранятся в ОЗУ. Это приводит к необходимости хранения программ на магнитофонных кассетах и в конечном счете создает определенные неудобства для пользователя. БИС РПЗУ серии К1601 позволяют строить одноплатные устройства памяти (до 32К байт). Причем нет необходи-

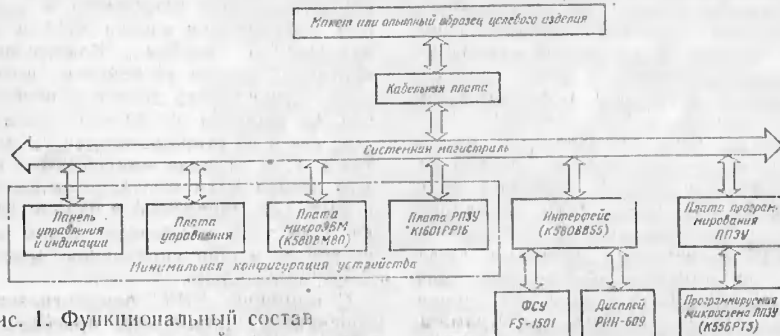


Рис. 1. Функциональный состав отладочного устройства

мости в отдельном устройстве программатора РПЗУ, так как интерфейс его схемы, реализующий все режимы работы РПЗУ, может быть размещен на одной плате с БИС РПЗУ.

После комплексной отладки микропроцессорной системы возникает задача прошивки программ в ППЗУ. Для этого можно использовать технические средства отладочного устройства: процессор и память. Добавление платы ключей и несложной программы прошивки позволяет программировать ППЗУ с минимальными аппаратными затратами.

Приведенные соображения по построению отладочного устройства реализованы в несложном приборе.

Минимальная конфигурация устройства (рис. 1) включает в себя па-

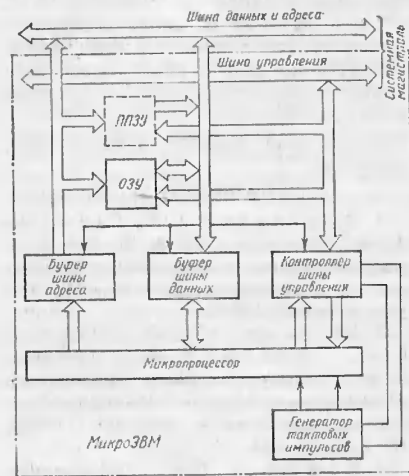


Рис. 2. Структурная схема микроЭВМ отладочного устройства

нель управления и индикации (лицевую панель) и три схемные платы размером 170×200 мм: плату управления, плату микроЭВМ, плату РПЗУ. Пользователь ведет диалог с отладочным устройством при помощи командной, режимной и двончной клавиатуры данных и светодиодных индикаторов лицевой панели.

Устройство отладки, кроме того, содержит средства, упрощающие эксплуатацию стенда и расширяющие область его применения. Подключение дисплея, например, дало возможность более оперативно вводить программы в шестнадцатеричной системе (режимы и команды по-прежнему должны задаваться с панели управления). Со встроенными средствами редактирования дисплея РИН-609 легко изменить отлаживаемую программу.

Проверенная программа под управлением микроЭВМ через плату программирования ППЗУ (рис. 1) прошивается в ППЗУ. Фотоэлектрическое устройство ввода с перфоленки позволяет вводить в РПЗУ объектные программы, полученные с помощью кросс-средств. Объем программы пользователя достигает нескольких

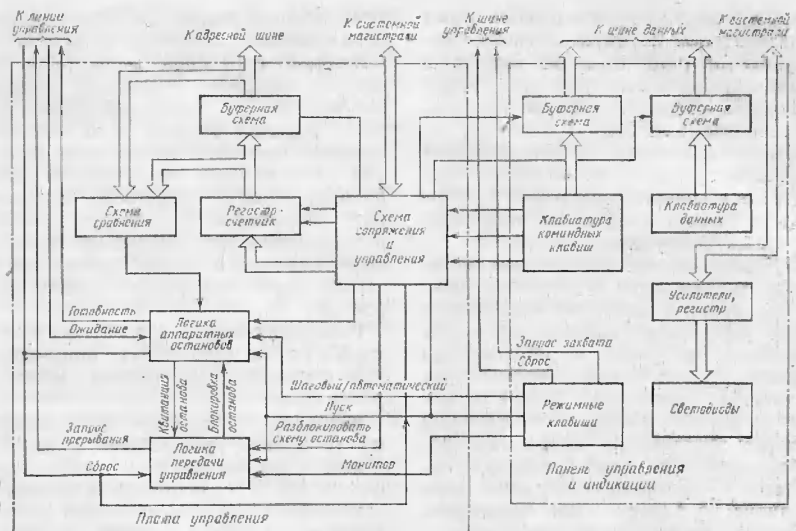


Рис. 3. Структурная организация платы управления

килобайт, так как, применяя технику программирования в машинных кодах, можно вносить изменения и в сложные программы [1].

Устройство отладки можно подсоединить к разъему микромашины целевого изделия кабелем (рис. 1) для совместной настройки программы и аппаратуры разрабатываемого изделия. Оборудование отладочного устройства в этом случае выполняет (эмулирует) все функции микроЭВМ проектируемого устройства. Для упрощения реализации данной функции все поле адресов микроЭВМ поделено между устройствами системы отладки и целевого изделия.

С панели управления можно просматривать и модифицировать регистры, ячейки устройств, подключенных к системной магистрали, устанавливать пошаговый режим выполнения программы, задавать точку останова, запускать программы с требуемого адреса. Пользователь вводит с дисплея свои программные модули в заданную им область памяти, выводит на экран дисплея и редактирует содержимое заданных областей памяти. Пользователю также доступны для просмотра и изменения регистры кристалла микропроцессора.

Простая микроЭВМ на базе микропроцессора К580ВМ80 — ядро рассматриваемой системы. На ее плате размещены модуль процессора, ОЗУ на 1024 8-разрядных слов и могут быть установлены два корпуса микросхем ППЗУ серии К556РТ5 (рис. 2).

Процессорный модуль состоит из микропроцессора, генератора тактовых импульсов, буферов шин данных и адреса, контроллера шины управления. В контроллер шины управления входят блок приоритетных прерываний, системный контроллер и схемы привязки входных сигналов микропроцессора: готовности, сброса, запроса захвата шин. Кроме того,

здесь же усиливаются выходные сигналы микропроцессора: ожидания, подтверждения захвата шин, разрешения прерывания.

ОЗУ реализовано на двух микросхемах К541РУ2Е. Шины микроЭВМ образуют системную магистраль, к которой подключаются все остальные устройства системы отладки.

Плата РПЗУ (8192 восьмиразрядных слов) построена на микросхемах серии К1601РР1Б. В эту память загружается программа пользователя либо под управлением микроЭВМ из ОЗУ дисплея, либо в режиме захвата шин с панели управления. В отсутствие всякого ПО последний режим использовался для ручного ввода системного монитора (дисплейного загрузчика, отладочного монитора, программ прошивки) в РПЗУ. Пользователь, нажимая соответствующие клавиши на лицевой панели, посылает необходимые сигналы магистрали (вырабатывает плата управления).

Плата управления принимает и исполняет команды от панели управления и микроЭВМ: устанавливает точку останова, инициирует переход микроЭВМ к выполнению программ монитора, организует пошаговый режим, ручной режим ввода-вывода в память стенда, останов по заданному адресу. Плата управления содержит схему сопряжения и управления, 16-разрядные регистр-счетчик и схему сравнения, логику аппаратных остановов и передачи управления (рис. 3).

Через схему сопряжения и управления микроЭВМ загружает в регистр-счетчик код адреса останова, передает команды на останов процессора (при обращении к клавиатуре данных) и на включение логики аппаратных остановов (при передаче управления отлаживаемой программе пользователя). Данная схема также организует режим захвата шин для

операций ввода-вывода, выполняемых пользователем вручную. Регистр-счетчик играет роль счетчика адреса и подключается к адресной шине через буферную схему, имеющую три состояния (см. рис. 3).

Схема сравнения обнаруживает равенство кода адресной шины магистрали с содержимым регистра-счетчика и выдает сигнал на логику аппаратных остановов.

Функции логики аппаратных остановов заключаются в переводе микропроцессора в состояние «ожидание» после получения сигнала от схемы сравнения, а также в организации пошагового (по циклам процессора) исполнения программ. После нажатия на панель режимной клавиши «Старт» выполнение программы возобновляется. Когда процессор находится в состоянии «ожидание» (при останове по адресу или пошаговом режиме), пользователь может по состоянию светодиодов лицевой панели судить о правильности выполнения программы и программного обмена микроЭВМ с устройствами магистрали.

Для более детального анализа необходимо зафиксировать режимную клавишу «Монитор», и тогда клавиша «Старт» передаст управление системному монитору, а логика аппаратных остановов отключится. Эту функцию реализует логика передачи управления (вырабатывает запрос на прерывание, вызывающий монитор). При возврате в отлаживаемую программу по команде монитора происходит разблокировка логики остановов.

На панели управления и индикации прибора находятся пять режимных и семь командных клавиш, во-

семь клавиш данных и светодиода, отражающие состояние отладочного монитора, системной магистрали и микропроцессора. Режимные клавиши производят системный сброс, задают режимы захвата и пошаговый, режимы работы с монитором. Функции этих клавиш реализуются только аппаратными средствами отладочного устройства. Клавиатуры данных и командная программно-доступны и подключаются к шине данных магистрали через буферную схему (см. рис. 3).

Подпрограммы отладочного монитора (256 байт), инициируемые с помощью командных клавиш, реализуют:

- передачу управления по заданному (с клавиатуры данных) адресу; вывод на индикацию ячейки (регистра УВВ) с заданным адресом;
- модификацию содержимого ячейки (регистра УВВ) с заданным адресом; просмотр содержимого регистров микропроцессора;
- модификацию содержимого регистров микропроцессора;
- установку кода адреса останова;
- возврат из монитора в отлаживаемую программу.

Сигналы трех из семи командных клавиш поступают также в схему сопряжения и управления (см. рис. 3), которая в ручном режиме ввода-вывода после нажатия клавиши «Старт» формирует сигналы на линии шины управления.

Устройство сопряжения дисплея и фотосчитывателя с магистралью подробно описано в работе [1]. Дисплейный загрузчик (≤ 200 байт) вводит набранную на дисплее информацию в установленную пользователем область памяти и выпол-

няет обратную задачу, причем по необходимости кодирует и компонуется передаваемую информацию.

Программа загрузки с перфоленты (64 байт) загружает объектную программу в заданную область памяти. Плата программирования (рис. 1) формирует (под управлением микроЭВМ) импульсы программирования с необходимыми длительностями и амплитудами, а также программно контролирует правильность прошивки ППЗУ. Программа прошивки (менее 250 байт) управляет ходом программирования и контроля.

Настольное устройство конструктивно оформлено (без дисплея) в корпусе с размерами $280 \times 480 \times 280$ мм; питается оно от пяти встроенных источников с напряжениями $+5$, $+12$, -12 , -30 , $+17$ В. Опыт эксплуатации устройства показал его эффективность на всех этапах проектирования микропроцессорной системы с ПО до 4К байт.

Телефон для справок: 7-54-23 (Махачкала).

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. — М.: Радио и связь, 1984.
2. Широков Ю. Ф., Осипов Е. Н., Жданов В. И., Гутовец Н. И. Учебно-отладочное устройство «Электроника-580». — Микропроцессорные средства и системы, 1985, № 2, с. 80—84.
3. Кофрон Дж. Технические средства микропроцессорных систем. — М.: Мир, 1983.

Статья поступила 15 декабря 1985 г.

КАК УЧИТЬ ПРОГРАММИРОВАНИЮ

С. П. Митрофанов

ПУТИ РЕШЕНИЯ ПРОБЛЕМЫ КОМПЬЮТЕРНОЙ ГРАМОТНОСТИ УЧАЩИХСЯ СЕВЕРНОЙ СЕЛЬСКОЙ ОПОРНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ СРЕДНЕЙ ШКОЛЫ

Сообщение на ВДНХ СССР от 16 апреля 1986 года

От редакции. В период принятия решения о повсеместном введении курса основ информатики и вычислительной техники в средней школе наибольший тревосгу вызывала необходимость учить информатике, не имея компьютера. Считалось, что школы, где есть хоть какая-нибудь вычислительная техника, уже имеют огромную фору, которой они смогут распорядиться.

Как ни странно, многие из таких «передовиков» оказались в затруднительном положении. Программа и учебник, ориентированные на отсутствие машин, не давали им прямого руководства к действию. Минпрос РСФСР передал инициативу им в руки, которой они не смогли воспользоваться. Недостаток нормативного материала привел к дезориентации и недовольству.

Как начать, как не потерять времени, как создать в школе обстановку нетерпеливого ожидания компьютера, как втянуть в работу шефов? Для многих эти вопросы оказались наиболее трудными.

Предлагаемый вниманию читателя материал дает, как нам кажется, убедительный ответ на указанные вопросы. Некоторые элементы стиля этой статьи, возможно, покажутся необычными для суховатого научно-технического журнала, заполненного схемами и текстами программ. Все-таки мы сочли необходимым напечатать эту статью во всей ее «непричесанности», потому что любой творческий труд — это труд с радостью, с восторгом от сознания приближенности к большому делу, ощущение движения и первых успехов. Мы хотели бы, чтобы каждый читатель увидел одновременно и энтузиазм, и профессиональную хватку автора из школы Тюменского Севера.

Общая характеристика школы. Опорная Сергинская средняя школа является единственной в поселке Сергинский Октябрьского района Тюменской области. В настоящем 1985/86 учебном году в ней обучается 1428 учащихся, работает 82 педагога, имеется 35 учебных кабинетов, мастерская по обработке древесины, цех по изготовлению школьной и детской мебели, комната ручного труда для начальных классов, автокласс, мастерская по металлообработке, кабинет обслуживающего труда и лаборатория информатики и вычислительной техники. Школьники имеют возможность обучаться профессиям столяра, швеи, шофера, воспитателя детского сада, парикмахера и оператора-программиста ЭВМ.

Пять базовых предприятий активно участвуют в решении проблем, связанных с реформой общеобразовательной и профессиональной школы, продолжая давно сложившиеся традиции совместной работы предприятий поселка и Сергинской средней школы по обучению, воспитанию и развитию всесторонней и гармонической личности каждого школьника.

Создание и оборудование лаборатории информатики и вычислительной техники. К возможности организации школьного вычислительного центра администрация Сергинской средней школы пришла еще в начале 1982/83 учебного года, получив материальную поддержку от шефов школы — Специализированного управления подводно-технических работ №10 (СУПТР-10) треста Сургут-подводтрубопроводстрой Миннефтегазстрой, которое в настоящее время является базовым предприятием школы, ответственно участвуя в организации процесса обучения учащихся профессии «оператор-программист ЭВМ». С 1 сентября 1982 г. началось обучение учащихся сельской средней школы программированию на языке Фортран с его графическим расширением. Практическая работа кружковцев проходила 2 раза в год — в зимние и летние каникулы. Для этого были организованы поездки в г. Свердловск, в Институт математики и механики УНЦ АН СССР, где ученики могли убедиться в правильности составленных ими программ, работая на машинах БЭСМ-6 и СМ-4. Но такие поездки, конечно, были неудобны по многим причинам. Юные программисты мечтали о своей машине, установленной прямо в школе.

Прошел первый год после реформы — и их надежды сбылись. 15 августа 1985 г. была организована лаборатория информатики и вычислительной техники, оснащенная вычислительной техникой. Площадь лаборатории 30 м².

Выполняя решение исполнительного комитета Сергинского совета на-

родных депутатов «О подготовке Сергинской средней школы к новому 1985/86 учебному году в условиях реформы школы», базовые предприятия школы перевели на счет сельского Совета материальные средства для приобретения микрокалькуляторов.

В лаборатории имеются программируемый микрокалькулятор БЗ-34 для работы кружка «Юный программист», 35 инженерных микрокалькуляторов типа БЗ-36 для уроков по основам информатики и вычислительной техники и 30 микрокалькуляторов типа МК-51 для быстрого получения ответов при решении задач вычислительного характера с реальными числовыми данными на уроках физики, химии и математики.

В лаборатории установлены две вычислительные управляющие микросистемы ВУМС-001, в состав каждой из которых входят: устройство специализированное управляющее вычислительное «Электроника ДЗ-28» 15ВМ32-021, блок сопряжения БС-02, дисплей 15ИЭ-00-013, термопечатающее устройство 15ВВП80-002. Завод, изготавливающий микросистемы, программное обеспечение для ВУМС-001 не поставляет. По нашей просьбе программное обеспечение для этих микросистем — язык Бейсик (версия 3А) — нам любезно предоставила кафедра вычислительной математики Свердловского государственного педагогического института.

С 1 сентября 1985 г. на факультативе для девятиклассников и в кружке «Юный программист» для 7—8 классов началось обучение языку программирования Бейсик.

Техническое обслуживание лаборатории производится высококвалифицированными специалистами из Института математики и механики УНЦ АН СССР. В лаборатории имеется система подготовки программ 15ИПГ, в состав которой входит следующее оборудование: «Электроника ДЗ-28» 15ВМ128-018, дисплей алфавитно-цифровой «Электроника» 15ИЭ200 × 140-017, термопечатающее устройство 15ВВП80-002, программное обеспечение — интерпретатор, Бейсик ДЗ-28 (версия 3А).

Получены два устройства внешней памяти СМ5211. В состав каждого устройства входят два кассетных накопителя на магнитной ленте. Подключение этих устройств к микросистемам ВУМС-001 даст нам возможность более эффективной и надежной организации ввода-вывода, библиотек и архивов данных.

Приобретен цветной телевизор «Горизонт 736». Цель использования телевизора как большого цветного видеомонитора состоит в выдаче на всеобщее обозрение визуальной текстово-графической информации, используемой в школе.

Для обслуживания и ремонта вычислительной техники приобретены

следующие приборы: осциллограф двухлучевой универсальный С1-74/1, осциллограф специальный С9-1, два осциллографа универсальных С1-73, два комбинированных прибора Ц4328.

Большое внимание уделяется сервизному оргтехническому оборудованию рабочего места, повышающему производительность труда оператора-программиста или референта. В лаборатории имеется четыре двухтумбовых стола. На каждом из них есть настольная лампа, пишущая машинка, подставка для книги, машинка для заточки карандашей, кювета для бумаг и специальный большой пенал со всеми необходимыми учебно-канцелярскими принадлежностями. Лаборатория достаточно обеспечивается канцелярскими товарами, предметами для делопроизводства, бумагой, лентой и копиркой для пишущих машин.

Воспитанники лаборатории — девятиклассники Латыпов Урал, Заволока Владимир и Нагибин Александр — ведущие программисты школы. Каждый из них ведет свое собственное конкретное компьютерное направление.

Латыпов У. занимается проблемами разработки программного обеспечения уроков математики для двух классов и отвечает за состояние библиотеки, программ, созданных для базового предприятия СУПТР-10 на производственном обучении специальности «оператор-программист ЭВМ».

Заволока В. с увлечением занимается игровыми программами, такими как «Лунолет-1», «Морской бой», «Крестики-нулики» и другими, разработанными в лаборатории. Он хранитель библиотеки программ факультатива по программированию.

Нагибин А. является консультантом кружков «Занимательный компьютер» для 4—6 кл. и «Юный программист» для 7—8 кл.

Каждый ведущий программист из числа старшекласников имеет свои именные кассеты и общую тетрадь, куда фиксирует все сделанные им программы и постоянно дополняет список каталога программ. Намечается создание электронного каталога.

В настоящее время используется более 30 кассет. Имеющиеся фонды будущей библиотеки превышают 90 магнитофонных кассет. Перспективы дальнейшего развития материально-технической базы: на период с 1986 по 1990 г. СУПТР-10 выделило лаборатории необходимые денежные средства. К началу 1987 г. мы ждем прибытия двух микроЭВМ «Искра-226» для создания автоматизированной системы управления Сергинской средней школы и АСУ СУПТР-10. Полученный опыт работы со старшекласниками позволяет надеяться, что поставленная задача будет выполнена.

Работа с периодической печатью и

литературой в лаборатории информатики и вычислительной техники. Много внимания уделяется работе с периодической печатью. Нас интересует информация о том, как в других районах нашей страны решается проблема компьютерной грамотности учащихся. Мы знаем о замечательных успехах новосибирских ученых и педагогов, о специально созданной системе программирования «Школьника» для применения ее в школах, об языках начального обучения программированию «Робик» и «Рапира». Очень интересен для нас опыт москвичей, свердловчан и ленинградцев, Латвийского и Казанского университетов.

Лаборатория выписывает ряд технических, научно-популярных и реферативных журналов: «Автоматика, телемеханика и вычислительная техника. ОИВ. Вычислительная техника. Выпуск сводного тома», «Микропроцессорные средства и системы», «Программирование», «Управляющие системы и машины», «Квант», «Математика в школе», «Физика в школе», «Наука и жизнь», «Техника — молодежи», «Юный техник» и «ВДНХ СССР». С июля 1986 г. в лаборатории будут поступать переводные американские журналы «В мире науки» и «Электроника», а также польский журнал по программированию «Компьютер для Вас».

Все материалы, опубликованные в этих журналах, просматриваются сначала заведующим лабораторией — учителем информатики, а потом вывешивается рекомендательный список статей для прочтения. Учащиеся читают с удовольствием, задают многочисленные вопросы.

Все газеты, поступающие в школьную библиотеку, просматриваются тремя референтами по информатике и их помощниками, готовятся соответствующие библиографические карточки, найденные статьи приносятся для ознакомления с их содержанием заведующему лабораторией один раз в неделю.

Имеется библиотека из более чем 250 книг по информатике, кибернетике, вычислительной и микропроцессорной технике (двойные экземпляры в указанное число не входят). За 1986 г. библиотека еще пополнится примерно 40 книгами. В этом нам помогут магазины и отделы «Книга-почтой», с которыми школа давно поддерживает связь.

Самым популярным изданием является книга Г. А. Звенигородского «Первые уроки программирования», таких книг у нас 15 экз. Все учащиеся, посещающие факультатив по информатике, прочитали эту книгу, познакомились с языком «Робик» заочно.

Книгой М. Н. Аршинова и Л. Е. Соловского «Коды и математика» ученики пользуются при декодировании зашифрованных сообщений, которые

приносят в лабораторию младшие школьники.

Книги А. Чакань «Что умеет карманная ЭВМ» и Я. К. Трохименко, Ф. Д. Любича «Микрокалькулятор. Ваш ход!» читаются учащимися при выполнении творческих работ, а также при перекодировании программ, данных в этих книгах на языке микрокалькулятора, — на язык программирования Бейсик.

Ведущие программисты читают книгу Ф. Бауэра и Г. Гооза «Информатика» как дополнительную литературу к действующему учебному пособию по информатике. Они ищут интересные задачи в книге Ф. Бауэра, Р. Гнаца и У. Хилл «Информатика. Задачи и решения». А в настоящее время они увлеченно готовятся к олимпиадам по программированию и тщательно изучают книгу А. Л. Брудно и Л. И. Каплан «Олимпиады по программированию для школьников». Перед ними поставлена цель: повысить квалификацию программиста, решив все задачи из этой книги самостоятельно.

Популярна также книга Уэрзелла «Этюды о программировании». Три интереснейших задачи из этой книги с увлечением программируются.

С нетерпением ждут «информатики» подключения периферийных устройств СМ 5211 к микросистемам ВУМС-001, чтобы создать тематические и алфавитные электронные каталоги всех книг, имеющихся у нас в лаборатории, всех заинтересовавших нас статей из периодики. Дети мечтают о начале создания банка библиографических данных по теме «Информатика и вычислительная техника». Референты лаборатории хотят готовить простые электронные рефераты для газетных и журнальных статей.

О преподавании предмета «Основы информатики и вычислительной техники». Эффективность работы учителя информатики самым тесным образом зависит от предыдущей многолетней работы его школьных коллег. Дело не только в том, что учащиеся девятих классов должны знать определенные научные факты. Они должны обладать хорошо развитым мышлением, а значит уметь логично излагать свои мысли грамотным и правильным языком. Но, как показывает опыт, речь учащихся нередко развита недостаточно, а порой и просто скудна. При такой ситуации работа учителя информатики заметно усложняется.

Перед нами поставлена трудная образовательная задача — обучение юношества работе с вычислительной техникой. Ее можно решить, если развивать у учащихся «операторное мышление» и правильную «алгоритмическую речь». Для решения этой ответственной задачи мы стремимся активизировать деятельность учащихся на уроках таким образом,

чтобы каждый старшеклассник имел возможность «наговорить» на уроке не менее одной минуты, чтобы каждый ученик был опрошен, смог высказаться, ответить пусть даже на очень простой вопрос. Чтобы каждый из девятиклассников ушел с урока информатики поработавшим с удовлетворением, показавшим приобретенные знания и умения своим товарищам и учителю, который даст объективную оценку его деятельности. Такую активизацию мы организуем, применяя дидактический прием обучения: комментирование цепочкой, когда каждому учащемуся дается возможность прокомментировать ту или иную часть алгоритма, найти в нем ошибки или недостатки, предложить для включения в алгоритм какую-либо строчку, верно объяснить ее написание, внимательно проанализировать смысл каждого элемента строки, проверить правильность структуры алгоритма, соответствие служебных слов номеру столбца, пользуясь в монологических ответах терминологией и понятиями, применяемыми при составлении алгоритмов. Строчки, которые диктуют учащиеся при составлении алгоритма, записываются учителем на доске. Если учитель видит ошибку, то он все равно записывает неверно продиктованную строку и не без волнения ждет, когда кто-то из его учащихся обнаружит ее. Время нахождение и исправление ошибки оптимально определяет сам учитель и, если она не найдена, сообщает о ней классу.

Так поставленная работа на уроке информатики дает определенные результаты, удовлетворяющие нас в настоящий момент, — в первый год изучения информатики в средних учебных заведениях. Многие алгоритмы, как, например, поиск данного элемента в линейной таблице, учащиеся составили самостоятельно, работая коллективно, по «цепочке». Порядок же ответов, т. е. кто и за кем отвечает, их продолжительность устанавливает учитель, исходя из особенностей класса, внимательно следя за тем, чтобы каждый ученик за урок ответил хотя бы один раз.

Большую часть уроков информатики мы начинаем с так называемого десятиминутного «Практикума на микрокалькуляторах», а остальные уроки — с десятиминутного обзора периодической печати и литературы по информатике и вычислительной технике. На уроке каждый учащийся имеет свой персональный микрокалькулятор и с увлечением работает на нем, решая упражнения из приложения 1 «Работа с калькулятором» из действующего учебного пособия, «Основы информатики и вычислительной техники. Ч. 1» (с. 82—90).

Первоначальное развитие навыков работы с калькулятором идет под

внимательным руководством учителя с показом приемов работы, а затем учащиеся с места громко и четко комментируют свои действия: называют клавиши, которые они нажимают, и говорят результат, высвечиваемый на индикаторе калькулятора. В первые четыре практикума эти результаты синхронно записываются учителем на доске, а в последующие — это уже не делается.

Целесообразность применения калькуляторов на уроках физики, химии и математики несомненная, поэтому партия инженерных калькуляторов передана в кабинеты естественно-математического цикла.

На конец 1985/86 учебного года запланирован 25-минутный практикум «Использование калькулятора для выполнения алгоритма Евклида». Здесь учителю помогают рекомендации приложения «Исполнение алгоритмов с помощью калькулятора» из методического пособия для учителей «Изучение основ информатики и вычислительной техники. Ч. 1» (с. 177—182).

Практикум на микрокалькуляторе БЗ-36 учащиеся считают «украшением» урока информатики. Что они скажут, когда на их столах будут установлены персональные компьютеры?

Очень популярны у наших учащихся обзоры периодической печати и литературы по информатике и вычислительной технике. Два таких обзора сделал учитель, а затем, когда в лаборатории были обучены три референта (в каждом девятом классе свой референт), обзоры стали делать они. Во время последнего обзора референты зачитали отрывок из фантастической повести Д. Бишофа «Военные игры», напечатанной в журнале «Ровесник» № 2 за 1986 год, рассказы о статьях, напечатанных в журнале «Радио» о программировании на языке Бейсик. После обзора учащиеся задают много вопросов. Их учебно-познавательная активность и интерес к предмету возрастают. Считается, что урок информатики пропустить нельзя ни в коем случае, там так интересно!

Работа всех алгоритмов, разбираемых на уроке и задаваемых для составления на дом, программируется на факультативе по информатике. Каждый десятиклассник имеет возможность внимательно проанализировать листинг программы, выведенный на термопечатающее устройство, и посмотреть, как работает алгоритм, закодированный на языке Бейсик. На магнитофонных кассетах создается школьная библиотека программ к действующему пробному учебному пособию «Основы информатики и вычислительной техники».

К 1 сентября 1986 г. в лаборатории будут подготовлены 12 консультантов, которые станут активными помощниками учителя в проведении

уроков по информатике в следующем учебном году в десятых классах.

Предполагается введение электронного журнала на уроках информатики со следующего 1986/87 учебного года. Мы ставим также перед собой такую задачу, чтобы каждый учащийся уходил с урока получив оценку, тем более, что занятия с микропроцессорной техникой будут проходить по группам, равным половине класса.

Работа факультатива «Программирование на Бейсике» для десятиклассников. Программа факультатива была составлена так, что через три месяца занятий (по два часа лекций в неделю) учащиеся теоретически освоили основные приемы программирования и все операторы языка Бейсик версии 3А. Расписание практической части факультатива было составлено так, что каждый программист имел возможность поработать на машине не менее двух часов в неделю. Учащиеся, у которых сразу же образовался стойкий интерес к программированию, находили возможность работать на машинах по 10 и более часов в неделю. Мы им не препятствовали, понимая, что они обрели новое для себя хобби. Машины для практических занятий включались в 15 ч и работали до 20 ч ежедневно.

На сегодняшний день результаты обучения программированию таковы: на «отлично» занимаются 3 чел., на «хорошо» — 6 чел., на «удовлетворительно» — 5, а у одного школьника интерес к программированию очень быстро пропал.

Под оценкой «отлично» мы подразумеваем такие знания и умения учащегося, когда перед ним ставится задача по программированию, и он совершенно самостоятельно составляет алгоритм на алгоритмическом языке «Информатика-9» и безошибочно кодирует его на языке Бейсик. Таких учеников принято называть у нас ведущими программистами.

Под оценкой «хорошо» мы имеем в виду такие знания и умения учащегося, когда он самостоятельно находит алгоритм и кодирует его, но в более медленном темпе, чем ведущие программисты, и один-два раза обращается за помощью к педагогу.

Под оценкой «удовлетворительно» мы называем такие знания и умения, когда учащийся с помощью учителя и консультантов составляет алгоритм, но совершенно самостоятельно кодирует его на языке Бейсик.

Обучение школьников ведется по препринту А. Д. Левина «Программирование на языке Бейсик», который был выпущен Институтом физики металлов УНЦ АН СССР, в г. Свердловске в 1985 г. тиражом в 100 экз. Книга эта написана хорошим, удобным для самостоятельного изучения, языком. В ней описывается язык

Бейсик версии 3А для машины «Электроника ДЗ-28».

После прохождения основного теоретико-практического курса у «информатиков» началась пора творчества. Трое ведущих программистов стали готовиться к олимпиадам по программированию, другие дошито и скрупулезно изучали программирование на калькуляторе БЗ-34, кто-то решал свои собственные задачи. Например, А. Проккопьев, из 9А кл., начал составлять придуманную им игру «Трансрапид-05».

Большим успехом у коллектива «информатиков» было создание интереснейшей компьютерной игры «Лунолет-1». Для этого была тщательно изучена статья канд. техн. наук М. Пухова «Истинная правда» из журнала «Техника — молодежи» № 6 за 1985 г. В ней программа игры закодирована на языке микрокалькулятора БЗ-34. Перед коллективом «информатиков» была поставлена задача изучить язык микрокалькулятора БЗ-34, разобраться в алгоритме программы и перекодировать программу М. Пухова на язык Бейсик. Каждый из «информатиков» внес в нее какое-то улучшение, сделал предложения по изменению структуры алгоритма, искал причины сбоев машины. Теперь эта программа работает как видеурок, позволяющий учащимся проанализировать и понять модель движения космического корабля в скользящем пространстве.

Продолжается работа по разработке других игровых программ, напечатанных в журнале «Техника — молодежи», и других источников. Программистами сочиняются сценарии личных компьютерных игр.

На факультативе сделаны: «Трансрапид-05», «Книга-почтой», «Библиографическая карточка», «Каталог», «Список по алфавиту», «Быстрая сортировка», начало «Поисково-информационной системы» и «Письмо другу». Идет упорная работа над сложнейшей программой для будущей АСУ школы — «Расписание уроков». Все творческие поиски учащиеся отражают в тетрадях, хранящихся в лаборатории. Мы начинаем обдумывать вопрос об единообразии написания программ и их правильного документирования по ГОСТу.

Некоторых программистов заинтересовал вопрос создания эффективных программ по быстрой сортировке. Для этого они изучают специальную сложную литературу. Во время такого изучения выяснилось, что нужно самостоятельное знакомство с такими языками программирования как Алгол-60 и Фортран-4. Для шести программистов школы такое знакомство не представляет особого труда.

Эффективность работы факультатива сдерживает только то, что вычислительных машин в школе очень ма-

ло — всего три. Энтузиазм многих «информатиков» такой, что они готовы сидеть за экраном дисплея все каникулы и все воскресные дни. Для снятия интеллектуального напряжения нам крайне необходима комната спортивно-аутогенной тренировки с хорошей фонотекой.

Большую роль педагогический коллектив Сергинской средней школы отводит поощрению учащихся, делающих замечательные успехи в изучении основ информатики и вычислительной техники. Для них были организованы три поездки в Свердловск в пединститут, чтобы поработать на ПК «Ямаха», и в ИММ УНЦ АН СССР для работы на графопостроителях фирмы «Бенсон». Всем лучшим программистам школы вручены памятные подарки за успехи в обучении информатике — комплекты книг по программированию, вычислительной технике и кибернетике.

С каждым учащимся, обучающимся на факультативе, один раз в неделю учитель проводит подробнейшее собеседование о том, что сделано, что удается и что нет, каковы планы на следующую неделю, что прочитано и как понято прочитанное, какую литературу будет читать и зачем. Мы стремимся давать такую объективную оценку интеллектуального труда ученика, чтобы у него формировались стимулы знать информатику и программирование еще лучше, чем сейчас. Мы убеждены, что именно такой работы требует от нас пятилетка ускорения и именно так должна проходить ориентация детей к профессиям, связанным с информатикой, электронно-вычислительной техникой и кибернетикой.

Об обучении профессии «оператор-программист ЭВМ». Базовое предприятие Сергинской средней школы — СУПТР-10 подписало договор со школой о подготовке специалистов по профессии «оператор-программист ЭВМ».

Этой профессии обучается 10 человек из числа тех, кто посещает факультатив и знает программирование на 4 и 5. Перед коллективом операторов-программистов, в состав которого входят три девушки и семь юношей, ставятся реальные, очень нужные производству задачи из опыта работы базового предприятия и других организаций поселка. Инженером или работником управления СУПТР-10 подробно и доступно на общем собрании операторов-программистов формулируется задача, подаваемая в текстовом и графическом виде.

Во время написания учащимися программы представитель от СУПТР-10 постоянно находится рядом с ними, наблюдает за процессом ее создания и отвечает на вопросы программистов. Первоначально ведущие программисты составляют подробнейший алгоритм, затем их товарищи ко-

дируют его на языке Бейсик, коллективно отлаживают программу и приглашают заведующего лабораторией и инженера от СУПТР-10 присутствовать при пуске сделанной программы. Первые две производственные программы «Объектная смета на переход газопровода через речку Амня» составлял учитель, или вернее, мастер производственного обучения. Все остальные программы при профессиональном обучении делаются учащимися с наибольшим коэффициентом самостоятельности. А так как это удается, то радости ребят нет границ, глаза их сияют, в лаборатории праздник! Живые, нужные программы для работы предприятия своей практической направленностью очень привлекают ребят. Воспитательная роль таких программ огромна. Они видят, что программирование это не «вещь в себе», а дело, необходимое для ускорения научно-технического прогресса.

Во время профессионального обучения созданы программы: «Переход газопровода через речку Сухонь, Анны и Малой Оби», «Расчет заработной платы для СУПТР-10», «Обработка наряда» и «Компьютерный экзаме́н по технике безопасности при проведении подводно-технических работ».

О работе кружка «Юный программист» для 7—8 классов. Кружок «Юный программист» с 1 октября 1985 г. посещает 15 человек. Сегодняшние восьмиклассники на следующий учебный год будут изучать основы информатики, поэтому были проведены эксперименты по самостоятельному изучению действующего пособия «Информатика-9» членами кружка. Двое из них, к нашему удивлению, успешно с этим справились. Они умеют составлять простейшие алгоритмы на школьном алгоритмическом языке.

Постоянными консультантами кружка являются программисты-девятиклассники. Они с удовольствием учат своих младших товарищей языку Бейсик,

Конечно, успехи юных программистов, посещающих кружок, не так заметны. И это можно объяснить: времени, чтобы получить удовольствие от общения с машиной, им выделяется лишь по 45 мин. для занятий за экраном дисплея.

Работа кружка «Занимательный компьютер» для 4—6 кл. В кружке «Занимательный компьютер» занимаются более 20 учащихся 4- и 5-х классов.

На занятиях кружка перед ними выступают программисты-девятиклассники и юные программисты. Детей знакомят с вычислительными микросистемами, популярно объясняют, как работает вычислительная машина. Референты рассказывают о применениях микропроцессорной техники.

Чаще всего занятие проходит в диалоге с машиной. Младший школьник приходит в лабораторию, садится за дисплей и начинает играть в увлекательную игру. А если и проиграет, не огорчается — компьютеру можно и проиграть.

Для этого кружка в лаборатории специально написаны игровые программы: «Генератор фантастических сюжетов», «Лунолет-1», «Лунолет-2», «Сделай сам себе планету» (все указанные 4 программы перекодированы из журнала «Техника — молодежи»), «Крестики-нолики», «Морской бой» и «Таблица умножения». Сейчас создается «Лунолет-3».

Планируется создание игр на историческую тематику, таких, например, как «Назови правильно дату». Пишется программа «Калах», «Удивительный генератор», «Управление работой промышленного предприятия» по книге Уэрзелла «Этюды о программировании».

Нам очень интересно создавать программное обучение самим. Мы наблюдаем интересный психологический эффект от поведения младшего школьника во время диалога с машиной. Ребенка невозможно оторвать от экрана дисплея! Проще кружок начать, чем его закончить! Статья поступила 23 июля 1986 г.

СПРАВОЧНАЯ ИНФОРМАЦИЯ

РЕПРОГРАММИРУЕМОЕ ПЗУ КМ558PP3

Интегральная микросхема КМ558PP3 представляет собой электрически репрограммируемое ПЗУ (ЭРПЗУ) информационной емкостью 65 536 бит с организацией 8К×8. Изготавливается по n-канальной МДП-технологии. Число элементов в схеме — 152 785. Предназначена для использования в качестве ПЗУ с многократной сменой информации в различных системах цифровой обработки данных. Выпускается в 28-выводном металлокерамическом корпусе типа 2121-28-6 (рис. 1). Назначение выводов показано в табл. 1,

Для управления микросхемой КМ558PP3 используются четыре сигнала: \overline{CE} , \overline{OE} , ERA, Ur. Адресные и информационные выходы в микросхеме разделены, однако их можно объединить для работы в мультиплексном режиме. Структурная схема ЭРПЗУ приведена на рис. 2.

Описание режимов работы

Режим чтения. (Временные параметры режимов работы приведены в табл. 2, временные диаграммы на рис. 3). Вначале микросхема переводится

Временные параметры режимов работы микросхемы KM558PP3

| Параметр | Обозначение | Норма | |
|--|-------------|-------|------|
| | | мин | макс |
| Время выхода из режима программирования, мкс | t_R | 100 | |
| Время сдвига сигнала выбора относительно сигнала адреса, нс | t_{AC} | 0 | |
| Время удержания сигнала адреса после подачи сигнала выбора, нс | t_{CA} | 120 | |
| Время выборки разрешения считывания, нс | t_{ODR} | | 140 |
| Время сдвига сигнала разрешения относительно сигнала выбора при считывании, нс | t_{COR} | 260 | |
| Время выборки адреса, нс | t_{ACC} | | 430 |
| Время удержания сигнала выбора после подачи сигнала разрешения, нс | t_{OCR} | 100 | |
| Длительность сигнала разрешения, нс | t_{OER} | 170 | |
| Время перехода информационных выходов в третье состояние, нс | t_{OD} | | 50 |
| Время восстановления при считывании, нс | t_{CCR} | 140 | |
| Время сдвига сигнала выбора относительно сигнала разрешения, нс | t_{OC} | 80 | |
| Длительность цикла считывания, нс | t_{RCV} | | 500 |
| Время сдвига сигнала выбора относительно сигнала программирования при записи, мкс | t_{PCW} | 20 | |
| Время сдвига сигнала разрешения относительно сигнала выбора при записи, нс | t_{COW} | 260 | |
| Время сдвига сигнала разрешения относительно сигнала входной информации при записи, нс | t_{DOW} | 50 | |
| Время удержания сигнала входной информации после подачи сигнала разрешения при записи, нс | t_{ODW} | 200 | |
| Длительность сигнала разрешения при записи, мкс | t_{OEw} | 0,17 | 4500 |
| Время записи, мс | t_w | 4,5 | 5,5 |
| Время восстановления при записи, мкс | t_{CCw} | 100 | |
| Время цикла записи, мс | t_{WC1} | 5,6 | |
| Время сдвига сигнала разрешения стирания относительно сигнала программирования при стирании, мкс | t_{PE} | 20 | |
| Время сдвига сигнала выбора относительно сигнала разрешения стирания, мкс | t_{EC} | 0 | |
| Время стирания, с | t_C | 19 | |
| Время сдвига окончания сигнала разрешения стирания относительно окончания сигнала выбора, с | t_{CE} | 2 | 21 |

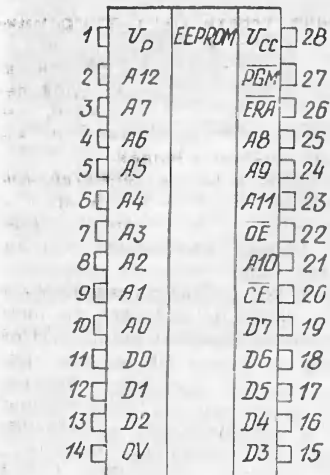


Рис. 1. Условное графическое изображение микросхемы KM558PP3

Таблица 1

Назначение выводов микросхемы ЭРПЗУ KM558PP3

| Вывод | Назначение |
|----------|--|
| 1 | Напряжение программирования, U_p |
| 2... 10 | Входы разрядов адреса, A12, A7... A0 |
| 11... 13 | Входы-выходы разрядов данных, D0... D2 |
| 14 | Общий вывод, 0V |
| 15... 19 | Входы-выходы разрядов данных, D3... D7 |
| 20 | Сигнал разрешения обращения, \overline{CE} |
| 21 | Вход десятого разряда адреса, A10 |
| 22 | Сигнал разрешения выдачи данных, OE |
| 23... 25 | Входы разрядов адреса, A11, A9, A8 |
| 26 | Сигнал разрешения стирания, ERA |
| 27 | Сигнал проверки программирования PGM |
| 28 | Напряжение источника питания, U_{cc} |

Таблица 3

Основные электрические параметры ЭРПЗУ KM558PP3, $T = -10... +70^\circ C$

| Параметр | Обозначение | Норма | | Параметр | Обозначение | Норма | |
|--|-------------|-------|------|--|---------------|-------|------|
| | | мин | макс | | | мин | макс |
| Напряжение низкого уровня. В при $I_{DOL} = 3.2$ мА | U_{DOL} | | 0,4 | Время выборки адреса, нс | t_{ACC} | | 430 |
| Напряжение высокого уровня. В при $I_{DON} = -0,4$ мА | U_{DON} | 2,4 | | Ток потребления от источника программирования при записи, мА | I_{PW} | | 10 |
| Ток потребления при обращении к микросхеме, мА | I_{CCA} | 60 | | Ток потребления при стирании, мА | I_{PE} | | 2,5 |
| Ток потребления при отсутствии обращения, мА | I_{CCS} | | 20 | Ток потребления при отсутствии обращения (пассивный режим), мА | I_{PS} | | 2,5 |
| Ток утечки на информационном выходе, мкА | I_{LDD} | -5 | 10 | Число циклов перезаписи | N_C | 100 | |
| Ток утечки по каждому входу: A0...A12, PGM, ERA, CE, OE, мкА | I_{L1} | -5 | 5 | Выходная емкость, пФ | C_o | | 22 |
| | | | | Входная емкость, пФ | C_i/C_{IOE} | | 3/5 |

УДК 62—83.001.7:621.314.6

В. М. Перельмутер

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ ЭЛЕКТРОПРИВОДАМИ

Микропроцессоры в системах управления современного автоматизированного электропривода увеличивают гибкость и приспособляемость к конкретным условиям технологического процесса, улучшают качество работы благодаря введению адаптации и самотестирования.

Автоматизированный регулируемый электропривод производственного механизма состоит из электродвигателя постоянного или переменного тока; источника его питания, преобразующего входное трехфазное переменное напряжение постоянной амплитуды и частоты в выходное напряжение постоянного тока, регулируемое по величине для питания двигателей постоянного тока, или в выходное напряжение переменного тока регулируемой частоты и амплитуды для питания двигателей переменного тока; системы регулирования основной и промежуточных координат (САР), системы управления режимами работы (СУ), системы защиты и сигнализации (СЗС).

В настоящее время в качестве источника питания используют различного рода тиристорные преобразователи (ТП). САР выполняют на аналоговых решающих и вычислительных элементах, цифровые элементы используют только для регулирования основной координаты, определяющей точность работы электропривода, в тех случаях, когда аналоговые регуляторы и измерители не обеспечивают требуемую точность. Системы СУ и СЗС выполняют с применением микросхем серий К155, К511, К561.

Технологический агрегат — это обычно механизмы (в том числе с электроприводами), связанные между собой технологией работы и конструкцией агрегата. В связи с этим и системы управления отдельных электроприводов связаны между собой с помощью системы управления режимами агрегата (СУРА).

В СУРА также применяют аналоговые элементы и микросхемы упомянутых серий; цифровые элементы используют при цифровом измерении и регулировании основной координаты электропривода, при необходимости в точном формировании

задающих сигналов в САР, для связи СУРА с ЭВМ более высокого уровня.

Микропроцессоры (МП) в рассматриваемых системах дают возможность уменьшить габаритные размеры и стоимость САР, СЗС, СУ, особенно в сложных модификациях электроприводов, повысить эффективность работы электроприводов за счет введения алгоритмов оптимизации и адаптации. Основное преимущество использования МП в электроприводах — возможность широкой унификации аппаратной части систем управления.

Микропроцессоры в электроприводах могут применяться [1] как для управления отдельным электроприводом (реализуя функции САР, СУ, СЗС), так и для управления группой взаимосвязанных электроприводов (агрегатом), т. е. в качестве элементов СУРА. Требования, предъявляемые к МП-системам, применяющимся в этих областях, различны. В дальнейшем будут рассматриваться электроприводы средней и большой мощности — от нескольких десятков кВт до 10... 12 тыс. кВт, изготавливаемые предприятиями электропромышленности для металлургической, горной, бумагоделательной и других отраслей народного хозяйства.

Микропроцессорные системы управления тиристорным электроприводом

Рассмотрим вначале характеристики микропроцессорных, т. е. МП-систем, реализующих функции прямого цифрового управления тиристорным электроприводом.

Разрядность. Разрядность МП-системы определяется требуемой точностью регулирования координат электропривода. Обычно достаточно точность 0,1%, так что с учетом знакового разряда возможно иметь 12-разрядное устройство. Учитывая, однако, необходимость в некоторых случаях (например, в системах позиционирования) достижения более высокой точности, а также существующую элементную базу, целесо-

образно принять число разрядов — 16.

Быстродействие. По своей физической природе ТП — дискретное устройство, так как сигнал управления на входе влияет на выходное напряжение только в дискретные моменты времени, соответствующие включению очередных тиристоров. Дискретность ТП в установившемся режиме $T_0 = 1/(mf)$, где m — число фаз выпрямления (обычно $m=6$), f — частота питающей сети (50 Гц) для ТП, ведомых сетью, или частота коммутации для автономных инверторов (где значение f достигает 60—75 Гц), т. е. $T_0 = 2,25—33$ мс. Для получения теоретически минимальной длительности переходного процесса в 1 такт T_0 , цикл вычислений регулирующего воздействия должен выполняться также за T_0 с некоторым запасом. За это время вводится цифровая и аналоговая информация о параметрах и координатах объекта управления, вычисляются сигналы обратных связей и входные сигналы регуляторов, различного рода компенсирующие воздействия, рассчитываются параметры регуляторов в контурах с адаптацией и заданные моменты подачи отпирающих импульсов на тиристоры, вводится ряд дискретных сигналов и определяется режим работы электропривода, вводятся и анализируются сигналы защиты электропривода, выводятся дискретные сигналы для включения вспомогательных цепей, сигнализации и т. п.

Как показывает программирование этих задач, в один цикл выполняется 0,5—1 К команд (типа регистр — память) из них 10—50 «длинных» команд типа умножения (деления). Таким образом, необходимо иметь время выполнения «короткой» команды порядка 1,5—2 мкс, а «длинной» — не более 8—12 мкс, что предъявляет достаточно высокие требования к быстродействию МП. В ряде случаев, например для некоторых типов электроприводов переменного тока, более выгодна установка двух МП, чем создание сверхбыстродействующей системы с одним МП.

В системе команд, кроме обычных операций (сдвигов, безусловных и условных переходов, перехода на подпрограмму и возврата из нее, арифметических, логических), для уменьшения длины программы, ускорения ее выполнения, облегчения программирования целесообразно предусмотреть специальные команды, как то: суммирование или

вычитание с ограничением; реализацию ПИ- или ПИД-регулятора; сравнение модулей чисел; формирование знака числа; вычисление функциональных зависимостей. Последние должны вычисляться табличным методом, желательно без интерполяции (требуемая точность относительно невелика). Необходимы как аналитические функциональные зависимости (обычно $\sin x$, $\cos x$, $\arcsin x$, \sqrt{x}), так и функции, получаемые экспериментальным путем (кривые намагничивания электродвигателей). Для реализации этих команд МП-система должна содержать ППЗУ, которые можно запрограммировать на месте эксплуатации.

Наиболее «быструю» прямую адресацию необходимо предусмотреть в первую очередь (наряду с остальными способами: косвенной, относительной).

Прерывание программ. МП-система управления электроприводом должна содержать иерархическую систему прерывания программ, причем обычно достаточно 4—5 уровней прерывания. В числе команд МП следует предусмотреть команды запрета и разрешения прерывания. Желателен также немаскируемый вход запроса прерывания, на который подается сигнал перезапуска от питающей сети или от таймера. Это резко уменьшает вероятность развития неконтролируемого процесса при сбоях в работе МП.

Память. Как показывает опыт программирования, для данной МП-системы достаточен объем ПЗУ 2 К байт и ОЗУ 1 К байт. При использовании большого числа таблиц объем ПЗУ должен быть увеличен.

Из периферийных устройств наибольшее внимание должно быть обращено на разработку преобразователя аналог (скорость, напряжение, ток) в код. Число аналоговых сигналов составляет 2...8 и более в зависимости от исполнения электропривода. Для получения приемлемых экономических показателей МП-системы целесообразно использовать один многоканальный ПНК, который последовательно преобразует в код аналоговые сигналы, в связи с чем он должен обладать достаточно высоким быстродействием. Следует остановиться еще на одном обстоятельстве. В электроприводах с двигателями переменного тока сигналы обратных связей (в установившемся режиме это постоянные величины) вычисляются МП на основании измерения некоторого количества аналоговых сигналов, являющихся синусоидальными функциями времени. Методическая ошибка в вычислении сигнала обратной связи не будет в том случае, когда синусоидальные функции зафиксированы в один и тот же момент времени. Пример — вычисле-

ние величин токов i_a , i_q во вращающейся системе координат по величинам фазных токов и угла между осью одной из фаз и положением вектора потокоцепления.

Если же зафиксировать эти величины с некоторым сдвигом, определяемым временем аналого-цифрового преобразования, то в сигналах обратных связей появится ошибка, исключить которую можно, либо введя аналоговые запоминающие устройства (которые в один и тот же момент времени запоминают синусоидальные сигналы, преобразуемые затем в код), либо усложняя алгоритмы обработки информации [2]. Радикальное решение вопроса — снизить время преобразования до 30—50 мкс (это уменьшает ошибку до допустимых значений).

Кроме ПНК, в составе МП-системы необходимо иметь устройства ввода дискретных сигналов (от 10 до 50 цепей, в зависимости от мощности и типа электропривода), а также вывода дискретных сигналов (5—15 цепей). Наиболее специфические — устройства, преобразующие выходной код МП-системы в последовательность импульсов управления тиристорами. Для ТП, связанных с питающей сетью, этот код в системе импульсно-фазового управления (СИФУ) преобразуется во временной сдвиг между моментом перехода через нуль сетевого напряжения и моментом подачи управляющего импульса; для автономных ТП этот код преобразуется в период следования импульсов управления. СИФУ состоит из узлов связи с питающей сетью и таймера.

Конструкция. Так как системы управления должны встраиваться в шкафы комплектных тиристорных электроприводов, конструкция которых определена отраслевыми стандартами на электрооборудование, то элементы МП-системы (вдвигные платы, блоки) должны также иметь конструкцию, определяемую этими стандартами. В частности, печатные платы должны иметь размер 233,4 × 160 мм или 233,4 × 220 мм с двумя штепсельными разъемами типа СМП-59.

Обслуживание. Число регулируемых электроприводов на металлургическом заводе составляет сотни и тысячи, обслуживает их персонал, как правило, с недостаточной квалификацией в цифровой вычислительной технике. В то же время требования к надежности достаточно высоки: время наработки на отказ — $(4-6,5) \cdot 10^3$ ч, время простоя при неисправности не более 10—15 мин. Увеличение надежности за счет дублирования, тройного резервирования с голосованием и т. п. существенно удорожает систему и делает ее неконкурентоспособной с аналоговыми системами.

Для увеличения надежности

должны быть максимально упрощены аппаратура и логика работы МП-системы, продублированы контактные соединения, введены самоконтроль с указанием отказавшей платы, использованы помехоустойчивые программы и др. Для облегчения поиска вышедших из строя ячеек необходимое испытательное оборудование должно быть включено в комплект поставки.

Отметим также, что тиристорный электропривод — это функционально законченное устройство и выпускается заводом-изготовителем с введенной в ПЗУ программой. Тем не менее, в процессе наладки на промышленном объекте в зависимости от технологии работы механизма может потребоваться внести изменения в программу работы МП-системы. Для этого необходимо предусмотреть определенные средства (например, установку перемычек). Кроме того, коэффициенты регуляторов, уровни ограничения и некоторые другие параметры системы управления определяются типом и характеристиками электродвигателя, особенностями механизма и другими факторами, поэтому необходимо предусмотреть возможность их достаточно плавного изменения при наладке, например с помощью потенциометров и ПНК. После наладки выбранные параметры должны быть переписаны в ПЗУ.

Пример. На основании изложенных выше требований институт ВНИИЭлектропривод [3, 4] была создана МП-система на базе микросхем серии К589, оперирующая с 16-разрядными словами. Длина микропрограммного слова — 24 разряда, тактовая частота 5 МГц. В системе имеется 32 команды типа либо аккумулятор — память, либо безадресные. Время выполнения команды — 2,2—3,2 мкс, команд умножения и деления — 10—25 мкс. Плата памяти содержит 1 К слов ППЗУ и 64 слова ОЗУ; можно установить две платы. МП-система связана с объектом с помощью плат ввода частотных, широтно-импульсных сигналов, цифровых кодов, непрерывных величин (ПНК типа напря-

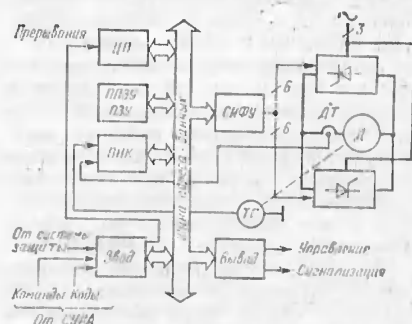


Рис. 1. Микропроцессорное устройство для управления электроприводами постоянного и переменного тока

жение — частота — код), а также плат вывода цифровой информации, плат связи с питающей сетью и с ТП. Эта МП-система, предназначенная для электроприводов постоянного тока мощностью от 10 до 1000 кВт, успешно прошла промышленные испытания [3].

В настоящее время этим же институтом совместно с НИИ ХЭМЗ разрабатывается модернизированное устройство с более широкими функциональными возможностями (рис. 1), предназначенное для электроприводов постоянного и переменного тока. Число команд увеличено до 50 (введены некоторые специальные команды), объем памяти увеличен (до 4К байт ПЗУ и 2К байт ОЗУ), введена многоуровневая система прерывания программ, добавлены косвенный и относительный способы адресации, существенно расширен состав периферийных ячеек, в частности, разработаны ячейки 16-канального 10-разрядного ПНК со временем преобразования 50 мкс, программируемого таймера, преобразователей частота — код и код — частота и др. Блок-схема типового алгоритма МП-системы изображена на рис. 2.

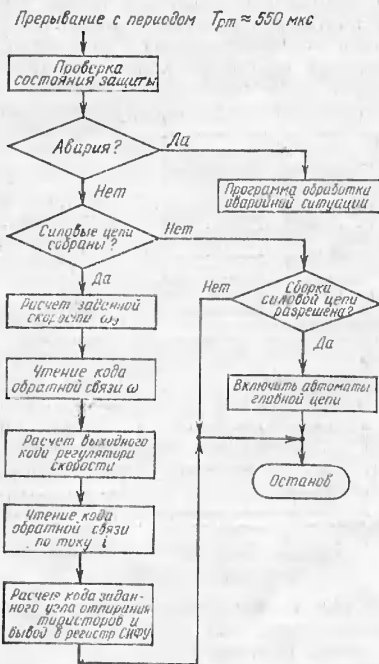


Рис. 2. Блок-схема алгоритма системы регулирования скорости электропривода постоянного тока

Микропроцессорные системы управления группой взаимосвязанных электроприводов (агрегатом)

Функции микропроцессорного устройства, используемого в СУРА,

определяются требованиями, предъявляемыми технологическим агрегатом.

Пример — СУРА для непрерывного 5-клетового стана холодной прокатки. СУРА получает около 200 дискретных (двухпозиционных) входных сигналов, определяющих режим работы, четыре кода, определяющих уровни скорости прокатки в различных режимах, два кода, определяющих темп разгона и торможения, 5 кодов диаметров прокатных валков, установленных в прокатных клетях, 4 кода, задающих соотношения скоростей прокатки в различных клетях (все коды двоично-десятичные, 8—12-разрядные) [4], аналоговые сигналы токов нагрузки и скоростей, электроприводов (12—14 цепей).

Выходные сигналы — примерно 30 двухпозиционных сигналов, поступающих в главные и вспомогательные электроприводы, а также пять кодов задания скоростей вращения электродвигателей клетей, которые рассчитываются по формуле [4]. Текущее значение заданной скорости (v_z) определяется процессом изменения с определенным ускорением или замедлением от установившейся скорости предыдущего режима до скорости заданного режима, причем этот процесс изменения может быть нелинейным, т. е. с переменным ускорением. При появлении перегрузки хотя бы одного из двигателей или превышения им допустимой скорости темп изменения v_z или значение v_z соответственно уменьшаются. Максимальное время цикла вычислений не должно превышать 5 мс [4]. Разрядность кодов v_z , i_1 , K_1 превышает 8, однако в СУРА применен и 8-разрядный процессор КР5801К80А, который может оперировать с 16-разрядными кодами. При этом необходима установка дополнительной аппаратуры: аппаратного умножителя, аппаратного преобразователя двоично-десятичного кода в двоичный (например, на микросхемах К155ПР6), устройств предварительной обработки входных сигналов (логические операции «И», «ИЛИ» над группами двухпозиционных сигналов, выделение наиболее-

го из групп аналоговых сигналов). Для рабочей программы достаточны объемы ПЗУ — 2...3 К байт, ОЗУ — 1 К байт.

Для СУРА можно использовать серийные микроЭВМ на базе БИС КР5801К80, например СМ-1800, дополненные аппаратным умножителем и некоторыми периферийными устройствами. В НИИ ХЭМЗ недавно разработано специализированное МП-устройство для рассматриваемого класса систем [4]. В его состав (рис. 3) входят центральная плата, плата памяти (ПЗУ — 3 К байт, ОЗУ — 1 К байт), аппаратный умножитель (на базе микросхемы К1802ВР2), программируемый таймер, устройства ввода и вывода цифровой и аналоговой информации. Для облегчения наладки системы в комплект поставки входит специальный стенд, в котором ПЗУ заменено на ОЗУ. В это ОЗУ можно вводить программы с перфоленты или магнитофона, корректировать программы и вносить отработанную программу в ПЗУ типа КР556РТ4 или К573РФ1. Конструкция устройства соответствует принятой в отрасли. Система успешно эксплуатируется на металлургическом заводе в составе СУРА стана холодной прокатки.

В заключение отметим, что для широкого внедрения микропроцессорных средств в автоматизированный электропривод необходимо решить не только инженерно-технические, но и, главное, организационные задачи, связанные с недостаточной квалификацией проектантов, наладочного и эксплуатационного персонала в цифровой вычислительной технике, в программировании. Без переподготовки этого персонала нельзя рассчитывать на массовое применение МП в электроприводе.

Адрес для справок: 310105, Харьков, НИИ ХЭМЗ.

ЛИТЕРАТУРА

1. Романов А. К. Микропроцессорная техника и автоматизация народного хозяйства. — Микропроцессорные средства и системы, 1984, № 1, с. 3—6.
2. Перельмутер В. М. Особенности применения микропроцессоров для управления частотно-регулируемыми асинхронными электроприводами на основе ИИЧ. — Электротехника, 1983, № 12, с. 32—35.
3. Слежановский О. В., Файнштейн В. Г., Файнштейн Э. Г. — Микропроцессорное управление тиристорным электроприводом. — Электричество, 1984, № 2, с. 33—37.
4. Перельмутер В. М., Соловьев А. К. Цифровые системы управления тиристорным электроприводом. — Киев: Техника, 1983. — 104 с.

Статья поступила 10 июля 1985 г.

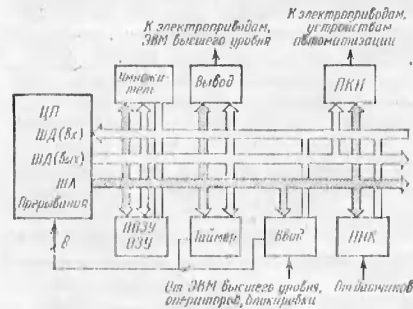


Рис. 3. Микропроцессорное устройство для управления агрегатом

СИСТЕМА АВТОМАТИЗАЦИИ ДИАГНОСТИКИ И КОНТРОЛЯ МСВТ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА МС 1212»

Повышение эффективности производства МСВТ связано с автоматизацией контрольных операций на основе автоматизированных внутрисхемных и функциональных комплексов контроля. Средства диагностики и контроля объединяются в локальную сеть звездной конфигурации под управлением рабочей станции. В пределах такой сети связываются автоматизированные комплексы внутрисхемного и функционального контроля с рядом программируемых терминалов или терминалов автоматизированных ремонтных комплексов. Среднемасштабный комплекс, использующий рабочую станцию уровня ввода, типа микроЭВМ «Электроника МС 1212», включает два автоматизированных комплекса контроля, два буквенно-цифровых программируемых терминала и один терминал автоматизированного ремонтного комплекса.

Число автоматизированных систем контроля в производственной линии определяется условиями производства, число автоматизированных комплексов внутрисхемного или функционального контроля — особенностями технологического процесса. Система диагностики и контроля МСВТ использует последовательную и параллельную организацию работ.

При последовательном контроле внутрисхемный комплекс предшествует функциональному. Внутрисхемный комплекс идентифицирует все неисправности, связанные с производственным процессом и компонентами для внесения быстрых исправлений, обеспечивает проверку выполненных исправлений перед подачей платы на функциональный контроль. Такая предварительная сортировка позволяет упростить программное обеспечение функционального комплекса, сконцентрировав его работу на обнаружении неудовлетворительных рабочих показателей, резко уменьшает дорогостоящее время для проведения диагностических операций.

При параллельном контроле функциональный комплекс первоначально решает задачу: годен/браков, что оптимизирует выход годных при первом проходе: отбракованные платы передаются внутрисхемному комплексу, годные — функциональному для проверки рабочих показателей. Платы с неудовлетворительными показателями направляются на ремонтную станцию с последующей проверкой внутрисхемным комплексом, а затем возвращаются в функциональ-

ный комплекс для повторной проверки.

В случае крупносерийного производства используется модифицированная схема параллельного контроля (рис. 1). Функциональный комплекс годен/браков обеспечивает оптимизацию выхода годных при первом проходе. Программирование для функционального испытания годен/



Рис. 1. Модифицированная схема параллельного контроля

браков является очень простым (отсутствует необходимость в каких-либо диагностических средствах); для диагностического функционального комплекса также упрощается, поскольку работа устройства ограничивается проверкой рабочих показателей и диагностикой. Основным преимуществом такой структуры является улучшение качества контроля и схемы технологического потока дефектных плат.

Автоматизированный внутрисхемный комплекс контроля (АВКК) с управляющей микроЭВМ «Электроника МС 1212» (рис. 2) обладает возможностями для тестирования активных и пассивных компонентов без отключения их от схемы, а также логических уровней цифровых интегральных схем, размещенных на печатных платах. Тестируемые платы устанавливаются в приемное устройство, представляющее контактный адаптер «ложе из гвоздей», позволяю-

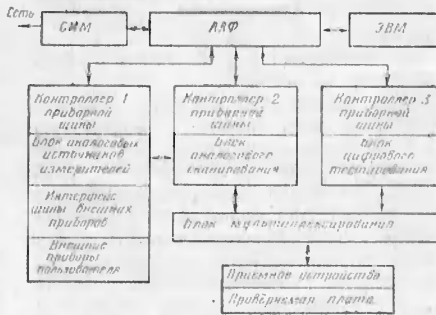


Рис. 2. Функциональная схема АВКК

щий АВКК (далее «станции») получить доступ к большому числу точек в схеме. Работа станции определяется следующими функциями: аналоговое тестирование, цифровое тестирование, обработка результатов тестирования, расширение контактного поля и работа интерфейса приемного устройства. Сетевой интерфейсный модуль (СИМ) позволяет подключить испытательную станцию к центральной рабочей станции системы.

Адаптер-адресный фильтр (ААФ) — буферное устройство, обеспечивающее передачу данных в направлении, определенном ЭВМ, генерирующее все управляющие сигналы, необходимые для обмена между шиной процессора и контроллерами приборной шины. Устройство ААФ может управлять тремя контроллерами приборной шины, которые выполняют в системе следующие функции: контроллер 1 управляет блоком аналоговых приемопередатчиков, 2 — блоком сканирования, 3 — работой блока цифрового тестирования и измерения. Контроллеры приборной шины адресуют приемопередатчики, управляя подключением их к контрольным точкам платы.

Мультиплексор (рис. 3) расширяет контактное поле, состоящее из 16 составных контактов для подключения к проверяемому устройству. Каждый составной контакт соответствует паре приемопередатчиков и одному контакту сканирующего уст-

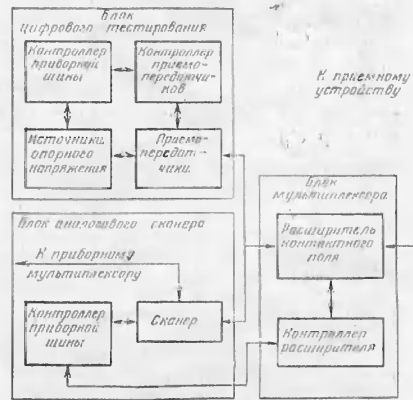


Рис. 3. Блок-схема системы мультиплексирования

ройства. Подключение ведется с помощью реле, образующих матрицы релейных контактов.

Приемное устройство служит механическим и электрическим интерфейсом между тестовой системой и адаптером. Может быть применена также вакуумная система (приемное устройство электрически соединяется с платами расширителя контактного поля, а при подаче вакуума игольчатые контакты адаптера замыкаются с контактами соединителя).

Во внутрисхемной станции используются две основные шины: общая шина процессора и приборная шина, по которой идет весь обмен данны-

ми между устройствами измерения и ЭВМ. После получения адреса устройства от ЭВМ ААФ выбирает один из контроллеров приборной шины и ретранслирует все последующие данные и сигналы управления этому контроллеру. Контроллер, в свою очередь, обменивается с адресованным устройством, вырабатывая сигналы синхронизации и управления, требуемые для обмена. Все устройства измерительной системы подключаются к этой приборной шине для осуществления программного управления.

При аналоговом тестировании контроллер осуществляет выбор устройства, его активизирование и генерацию векторов прерывания в режиме прерывания от ВУ. Во время обмена контроллер буферизует данные, поступающие к устройству и исходящие от него и вырабатывает все управляющие сигналы, необходимые для передачи данных по приборной шине. Под управлением контроллера (рис. 4) выводы приемопередатчиков подключаются через контакты реле к любой линии (А...Д) приборного мультиплексора, которая выводится к сканирующей системе для подключения к мультиплексору — расширителю контактного поля. Мультиплексор 1 предназначен для приборов стандартного набора, который включает:

источник тока (уровень устанавливается программно, диапазон регулирования, шаг квантования и точность поддержания значения тока соответствует требуемой точности измерений);

источник напряжения, требования к которому аналогичны изложенным выше;

измеритель тока, обеспечивающий измерение токов обих полярностей (при измерении тока величина последнего преобразуется в напряжение, диапазон измерений устанавливается программно);

измеритель напряжения, преобразующий напряжение в цифровую форму с помощью аналого-цифрового преобразователя. Полученное слово данных совместно с величиной установленного диапазона определяет значение измеряемой величины и может быть считано процессором;

измеритель импеданса по переменному току (импеданс измеряется путем подачи на тестируемый элемент переменного тока с последующим измерением синусоиды и косинусоид составляющих тока и напряжения). Результаты измерений в цифровой форме поступают в ЭВМ. Мультиплексор 2 предназначен для дополнительных источников питания, совместимых с вычислительными системами. Каждая линия мультиплексора сканируется приемным устройством, что позволяет производить все типы соединений, включая соединения с организацией обходных токов, изолирующие отдельные компоненты от остальной схемы. Наличие четырех мультиплексируемых линий обеспечивает возможность реализации четырехпроводной схемы измерения по системе Кельвина.

При цифровом тестировании ИС проверяемого устройства используются контакты приемопередатчиков. Напряжения логических уровней подаются на выбранные выводы ИС, анализируются выходные сигналы и их зависимость от времени.

При проведении тестов компонентов, установленных в схеме, каждая из них проверяется так, как если бы она была отключена, даже в тех случаях, когда на ее входы поступают сигналы с других схем проверяемой платы. Для этого источники сигналов должны обеспечивать большой выходной ток при низком сопротивлении источника, чтобы подавленные сигналы, приходящие от других ИС. Такой способ изоляции связан с перегрузкой выходов соседних ИС и повышенным выделением тепла на них, поэтому, чтобы не повредить элементы, последовательность тестов проводится за небольшое время. Таким образом, устройство для проверки компонентов без отключения их от схемы должно обладать высокой нагрузочной способностью по току и некоторой минимальной скоростью проведения тестов для предотвращения повреждения схем проверяемого устройства из-за чрезмерного рассеяния мощности в последних.

Цифровая подсистема (рис. 5) включает в себя:

приемопередатчики логических уров-

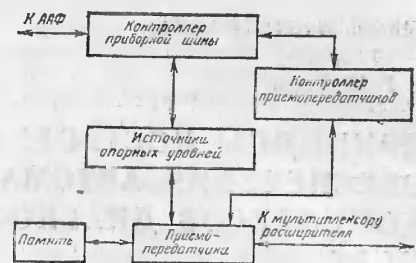


Рис. 5. Функциональная схема блока цифрового тестирования

ней сигнала для создания тестовых последовательностей и измерения отклика схем;

программируемые источники опорного напряжения для формирования логических уровней и порогов срабатывания логических элементов;

память для хранения последовательности входных сигналов и образцовых выходных последовательностей, с которыми сравниваются полученные от проверяемого устройства последовательности. Результаты позволяют судить об исправности проверяемого устройства;

контроллер управления последовательностью тестов и проведением тестирования;

интерфейс ЭВМ, обеспечивающий возможность записи и считывания данных ЭВМ и управления последовательностью выполнения тестов, в данном случае — контроллер приборной шины и ААФ.

Каждый цифровой контакт системы цифрового тестирования имеет собственную пару: приемопередатчик и память с произвольным доступом (ОЗУ), позволяющую быстро выполнять тестирование независимо от ЭВМ и обмениваться с ним пакетами по окончании тестирования.

Контроллер приемопередатчиков задает временную последовательность операций, дает возможность проводить проверку выполнения условий во время тестирования, координирует обмен данными между приемопередатчиками и ЭВМ, осуществляет адресацию устройств памяти и управления, соответствующих контактным штифтам, осуществляет шинные соединения, сбор данных с приемных схем. Связь между приемопередатчиками, соответствующим контроллером и источниками опорных уровней ведется по шине приемопередатчиков. По ней передаются опорные уровни, адресная информация, управляющие сигналы от контроллера к приемопередатчикам.

Источники цифровой системы тестирования — это устройства с тремя выходными состояниями: высокий уровень, низкий уровень, состояние высокого сопротивления. Приемники — двухпороговые приемные устройства для идентификации высокого, низкого, промежуточного уровня. *Статья поступила 17 декабря 1985 г.*

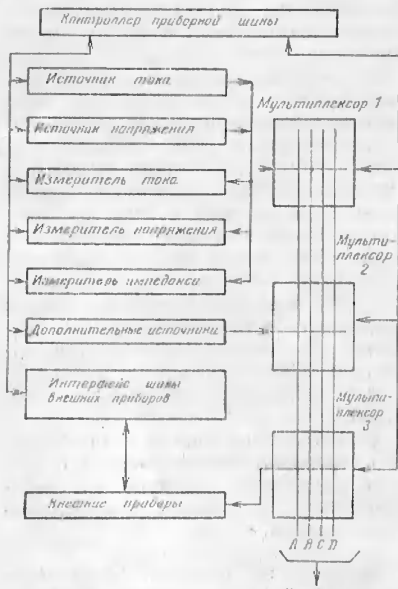


Рис. 4. Функциональная схема системы аналоговых источников измерителей

А. Г. Юрочкин

ПРИНЦИПЫ ПОСТРОЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННЫХ КОМПЛЕКСОВ ДИАГНОСТИКИ И КОНТРОЛЯ МСВТ

Программное обеспечение автоматизированных комплексов контроля (АВКК) должно обеспечивать внутрисхемное тестирование аналоговых компонентов и цифровых интегральных схем. При аналоговом тестировании программные средства должны изолировать или «защищать» испытуемый компонент и учитывать наличие сопротивления тест-гвоздей, реле и монтажа между печатной платой и измерительными инструментами. Сначала проводится проверка на связность, определяется наличие коротких замыканий и обрывов электрических связей на плате, измеряется сопротивление между возможными парами узлов. Несоединенные узлы тестируются на наличие высокого, а соединенные — низкого сопротивления.

Основные конфигурации импедансных тестов — это двухточечное незащищенное измерение и четырехточечное защищенное измерение. Двухточечное тестирование выполняется при подаче известного напряжения на компонент от программируемого источника напряжения и измерении тока на «виртуальной земле» амперметра. Четырехточечное измерение использует технику электрической изоляции для устранения шунтов. Тестируемый компонент защищается подачей напряжения (равного напряжению измеряемого узла) на все соседние узлы для предотвращения утечки тока через шунты.

При цифровом тестировании необходимо изолировать тестируемое устройство от окружающих схем подачей на короткое время логических импульсов большой силы тока. Тест устройства должен содержать наборы известных логических импульсов (подаваемых на вход ИС) и предпологаемых выходных состояний, т. е. таблицу истинности.

Рабочие программы внутрисхемного контроля строятся на основе библиотек аналоговых и цифровых тестов. Библиотеки организованы по иерархическому принципу и имеют файловую структуру. Библиотечные тесты сгруппированы в модули по типам цифровых устройств, а модули, в свою очередь, объединены в файл, поэтому доступ к тестам осуществляется при вызове имени файла и имени отдельного модуля. *Системный файл* содержит наиболее полный ком-

плект тестов и процедур существующих элементов. В этот файл запрещено внесение каких-либо изменений, — он является постоянной частью библиотек. *Модификационный файл* состоит из тестов и описаний на новые компоненты, предусматривает возможность модификаций существующих тестов, стирание файлов и запись новых. *Пользовательский файл* создается пользователем для каждого типа тестируемых устройств, содержит тест-процедуры на неизвестные компоненты, а также тесты всех устройств (аналоговых и цифровых) испытуемой платы. Наличие этого файла значительно сокращает время на поиск нужного теста.

Тест сложного цифрового устройства состоит из нескольких секций: «заголовок», «объявить», «отключить», «запретить», «вставить», «раздел тестирования шины», «специальная», «основная», каждая из которых последовательно описывает функциональную схему выбранной ИС. Каждый шаг теста является как бы перечнем состояний всех выводов ИС. В секцию «заголовок» входят число и перечень выводов, которые могут быть входами, находиться в третьем состоянии или быть неуправляемыми. В секции «объявить» определяются переменные, используемые в тесте. Секция «отключить» используется для разделения выводов ИС и шины. Секция «запретить» реализует способы защиты ИС от изменения выходного состояния во время тестирования соседней ИС. Процедура запрета сообщает, что нужно сделать, чтобы сохранить ИС в стабильном состоянии. Секция «вставить» определяет основные повторяемые тестовые операции.

Секция «специальная» необходима при замене сложной процедуры тестирования на более простую. Секция «основная», содержащая фактический тест, позволяет убедиться в том, что каждый функциональный вывод может устанавливаться в высокое или низкое состояние так, чтобы можно было проверить правильное соединение узлов на печатной плате. Для тестирования микропроцессора необходимо обеспечить синхронизацию циклов команд при отключенном синхрогенераторе на печатной плате. Тест «микропроцессор» переключает выводы высокого и низкого состоя-

ний, проверяет основные функциональные блоки, адресные режимы, типы данных (байт, слово), циклы синхронизации, выполняет типичные последовательности команд пользователя: чтение, запись, выборка и т. д.

При написании теста учитывают, что функциональный режим или режим адресации прямо соответствует биту в команде, поэтому можно формировать набор соответствующих тестовых команд, имея битовый набор и очищая его. После этого необходимо убедиться в том, что каждая функциональная единица и каждая магистраль данных тестируются. Кроме того, должны тестироваться прерывания и проверяться флаги состояний. Подготовку можно вести в режиме самоотключения с колодкой для установки ИС, используя логическую плату, которая непосредственно подключается к автоматизированному испытательному комплексу. Плата обеспечивает внутренние связи и соединения выводов данного устройства с соответствующими тестовыми точками системы. Тест-процедура генерируется под управлением системы (при подаче различных последовательностей автоматически изучаются выходные состояния). Еще более удобен метод модификации тестов, т. е. копирование уже существующего теста и его модификация с помощью команд редактирования в соответствии с требуемым элементом. Во время генерации рабочей программы доступ к библиотекам осуществляется автоматически в режиме меню. При этом сначала просматривается пользовательский файл на наличие нужного теста. Если тест не найден, просматривается модификационный файл, а затем системный. Если тест не обнаружен и в системном файле, то формируется соответствующее сообщение и система выходит в ожидание ввода следующей директивы. Если же тест найден, то вводится номера тест-точек для каждого вывода ИС. Перед вводом в рабочую программу библиотечный тест редактируется и автоматически вносится в определенную секцию программы. Система управления библиотеками позволяет использовать как пакетный, так и интерактивный режимы доступа.

Рассмотренные состав и структура ПО позволяют автоматизировать процесс разработки рабочих программ контроля с использованием единой базы данных.

Телефон для справок: 56-48-30 (с 20 до 21 часа), г. Воронеж, Юрочкин Анатолий Геннадьевич

Статья поступила 17 декабря 1985 г.

Н. Н. Щелкунов, А. П. Дианов

УНИВЕРСАЛЬНЫЙ ОДНОПЛАТНЫЙ МИКРОКОНТРОЛЛЕР

Микроконтроллер мМС1204 представляет собой законченную одно-платную микросистему общего назначения с магистрално-модульной архитектурой открытого типа (рис. 1). Базовый набор программно-аппаратных средств микроконтроллера (МК) обладает функциональной завершенностью и обеспечивает широкий круг его применения в качестве информационного ядра систем управления реального времени. В данной статье приводятся обоснование выбора и подробное описание аппаратных средств МК. Программные средства МК будут изложены в отдельной статье.

В состав микроконтроллера входят 8-разрядный центральный процессор (ЦП), ПЗУ объемом 4К байта, ОЗУ объемом 1К байт, два последовательных интерфейса ИРПС и параллельный интерфейс ИРПР. Системный таймер и 8-уровневая система прерываний обеспечивают поддержку режима реального времени. Внутрисистемная магистраль допускает многоплатные расширения микроконтроллера с целью увеличения его функциональных возможностей.

Центральный процессор. Полная принципиальная схема микроконтроллера приведена на рис. 2. В состав ЦП кроме микропроцессора КР580ВМ80 входят генератор тактовых импульсов КР580ГФ24 и системный контроллер КР580ВК28/ВК38. Период следования синхросигналов Ф1, Ф2 микропроцессора КР580ВМ80 определяется кварцевым резонатором Z1. Частота его колебаний должна лежать в диапазоне 4,5...22,5 МГц и быть в 9 раз больше желаемой частоты работы МП. При выборе и установке квар-

ца следует обеспечить минимальное последовательное сопротивление внешней цепи генератора XTAL2 — XTAL1. Со стороны кварца генератор представляет собой индуктивный элемент, приводящий к снижению рабочей частоты колебаний. Для компенсации данного эффекта рекомендуется последовательно с кварцевым резонатором со стороны вывода XTAL2 включить конденсатор небольшой емкости. Этот же конденсатор предохраняет кварц от смещения и повреждения в результате протекания через него постоянного тока. Для кристалла 18 МГц величина емкости лежит в пределах 15...30 пФ. Повышение частоты может потребовать уменьшения емкости конденсатора.

Кроме тактовых последовательностей Ф1, Ф2 генератор формирует системную тактовую частоту CCLK (ток нагрузки $I_{OL} \leq 15$ мА) и сигнал системного сброса INIT ($I_{OL} \leq 15$ мА). Сигнал INIT предназначен для установки системы в исходное состояние. Он формируется при включении питания и нажатии специальной клавиши RESET. Длительность сигнала определяется RC-цепочкой на входе RESIN генератора. Используемые в МК элементы обеспечивают длительность сигнала сброса, равную 0,35 с.

Сигнал подтверждения ХАСК используется при работе с медленными устройствами памяти и ввода-вывода. Он может быть задействован также для организации ожидания ЦП некоторого внешнего события. Микропроцессор КР580ВМ80 будет переведен в состояние ожидания WAIT, если сигнал ХАСК окажется неактивным в момент первой его

проверки внутри текущего машинного цикла (начало второй трети такта T2). В зависимости от размера и характеристик системы сигнал подтверждения может быть организован одним из двух способов.

В больших системах линия подтверждения ХАСК обычно выполняется с низким уровнем активности, что обеспечивает асинхронный способ доступа к устройствам памяти и ввода-вывода. Для этого перед входом RDYIN генератора следует поставить дополнительный инвертор. После того как выбранное устройство получило команду, оно генерирует сигнал подтверждения, формируемый на линии ХАСК по схеме с открытым коллектором. Для максимальной производительности системы ее модули должны возвращать сигнал подтверждения до его первой проверки.

В малых системах с быстрыми устройствами рекомендуется использовать линию ХАСК с высоким уровнем активности. По умолчанию ЦП работает в синхронном режиме с максимальной для него скоростью. Следовательно, в устройствах, успевающих выполнить операцию в синхронном режиме, можно опустить логику подтверждения обменом, тем самым упростив их.

В МК использован системный контроллер КР580ВК38. От контроллера КР580ВК28 он отличается тем, что формирует стробы I/OW и MEMW с упрощающим расширением. В микросхеме КР580ВК28 эти стробы представляют собой задержанный на 45 нс строб WR микропроцессора КР580ВМ80, который генерируется в такте T3, т. е. уже после первой проверки сигнала подтверждения ХАСК. Начало всех командных стробов КР580ВК38 привязано к стробу STSTB и задержано по отношению к его началу на 20...60 нс (рис. 3). Так как STSTB опережает на T/3 фазу Ф2 такта T2, то до момента первой проверки сигнала ХАСК еще остается (4T/9—60) нс. При T=400 нс оно составляет около 120 нс, что вполне достаточно для сброса сигнала ХАСК.

Применение контроллера КР580ВК28 допускается в системах с синхронным обменом или в системах асинхронного доступа с инверсной линией ХАСК. В микросистемах, допускающих оба протокола обмена, рекомендуется использовать микросхему КР580ВК38. Во всех случаях нагрузочная способность командных линий MRDC, MWTC, IORC, IOWC следующая: $I_{OL} \leq 10$ мА, $C_L \leq 100$ пФ. Допустимый ток I_{OL} и емкость C_L линии INTA равны 5 мА и 100 пФ соответственно. Системный контроллер обеспечивает также двунаправлен-

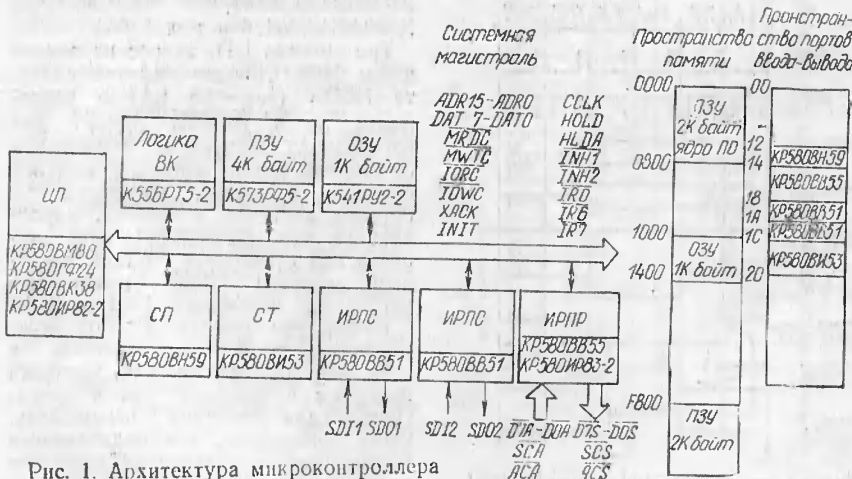


Рис. 1. Архитектура микроконтроллера

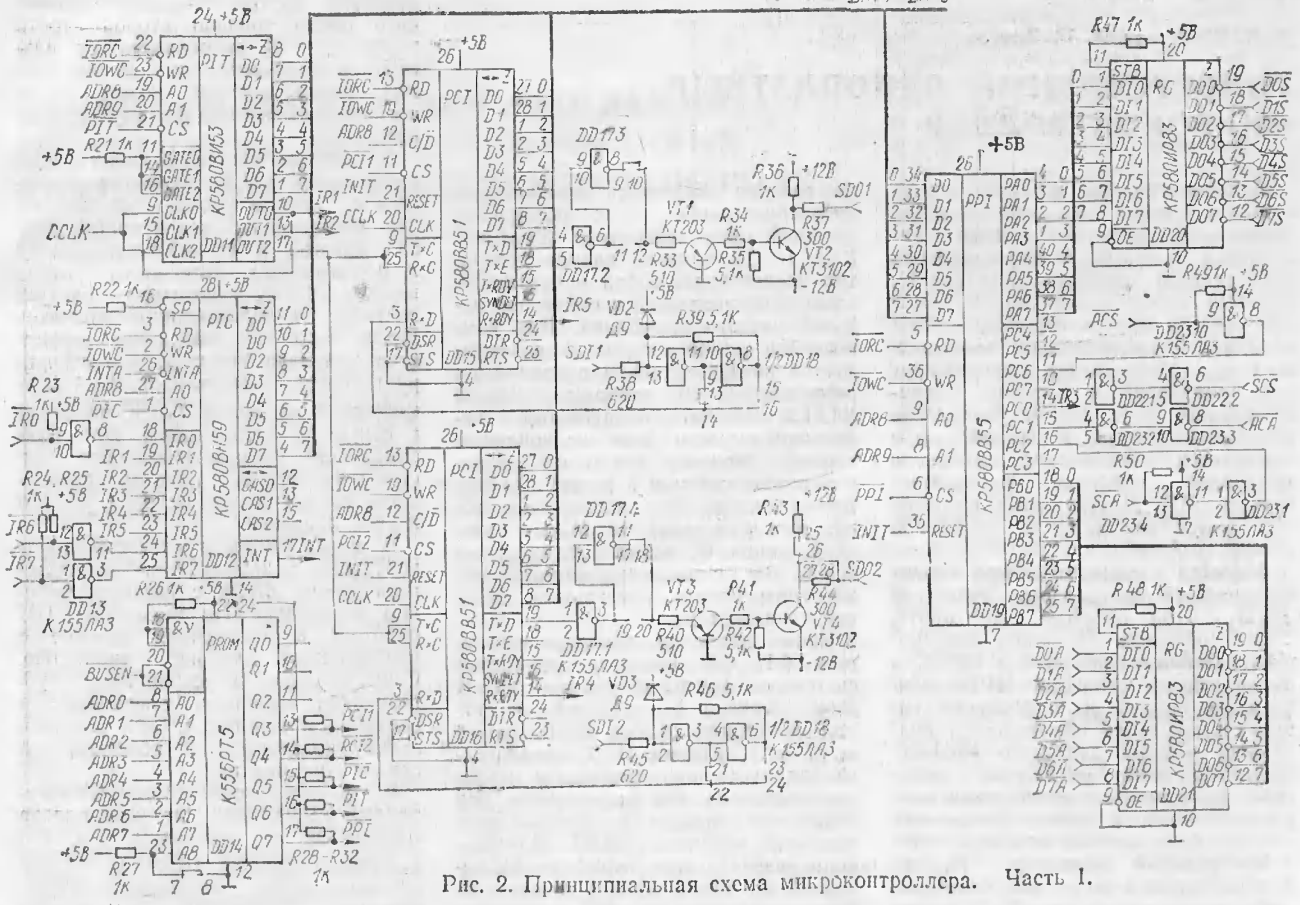


Рис. 2. Принципиальная схема микроконтроллера. Часть 1.

ную буферизацию шины данных МП без инверсии, доводя ее нагрузочную способность по току I_{OL} до 10 мА и емкости C_L до 100 пФ. Дополнительная задержка в канале данных составляет около 30 нс. Удобно, но не обязательно, системную шину данных DAT7...DAT0 нагрузить на +5В резисторами в 10 кОм. Это практически не скажется на нагрузочных свойствах шины,

но чтение команды из несуществующей области памяти приведет к выполнению команды RST7, что может быть использовано в отладочных и диагностических целях. Адресная шина микропроцессора KP580BM80 обеспечивает токовую нагрузку $I_{OL} \leq 1,8$ мА и емкость нагрузки C_L до 100 пФ. Этой нагрузочной способности может быть достаточно для построения одноплат-

ного МК закрытого типа. Принцип открытости структуры потребовал буферизации адресной шины с помощью двух буферных регистров KP580IP82 без инверсии. Буферирование адреса позволило увеличить максимальные значения I_{OL} до 32 мА и C_L до 300 пФ. Вместе с этим в шине появилась дополнительная задержка в 35 нс. Возможна реализация адресного буфера на других схемах, например: KP580IP83, KP580BA86/BA87, KP589IP12, K589AP16/AP26 и т. д.

Три сигнала ЦП: запрос на захват шины HOLD, подтверждение захвата HLDA ($I_{OL} = 1,8$ мА) и разрешение шины BUSEN служат для организации доступа к магистрали со стороны других активных модулей системы, например, устройства с прямым доступом к памяти, управляемого БИС KP580BT57. Для запуска ЦП в системах с одним активным модулем (ЦП) необходимо установить перемычки 1-2 и 3-4.

Подсистема памяти. Память микроконтроллера должна включать в себя как постоянную для хранения программ и констант, так и оперативную для хранения переменных. Опыт показывает, что подавляющее большинство управляющих программ занимает область около 4К байт и

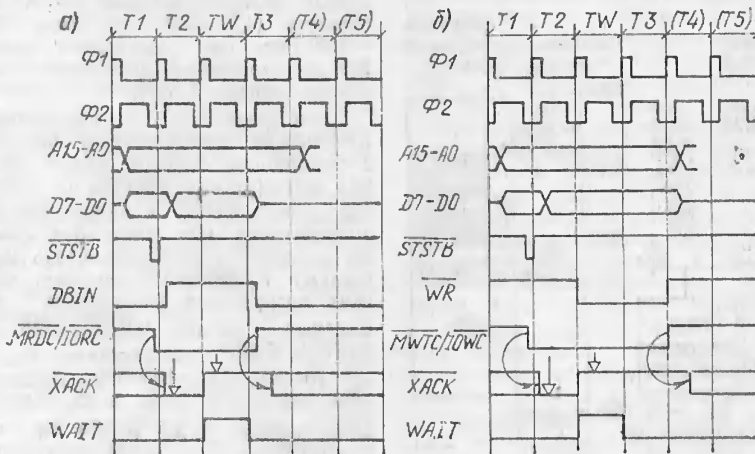


Рис. 3. Временные диаграммы работ ЦП: цикл ввода (а), цикл вывода (б)

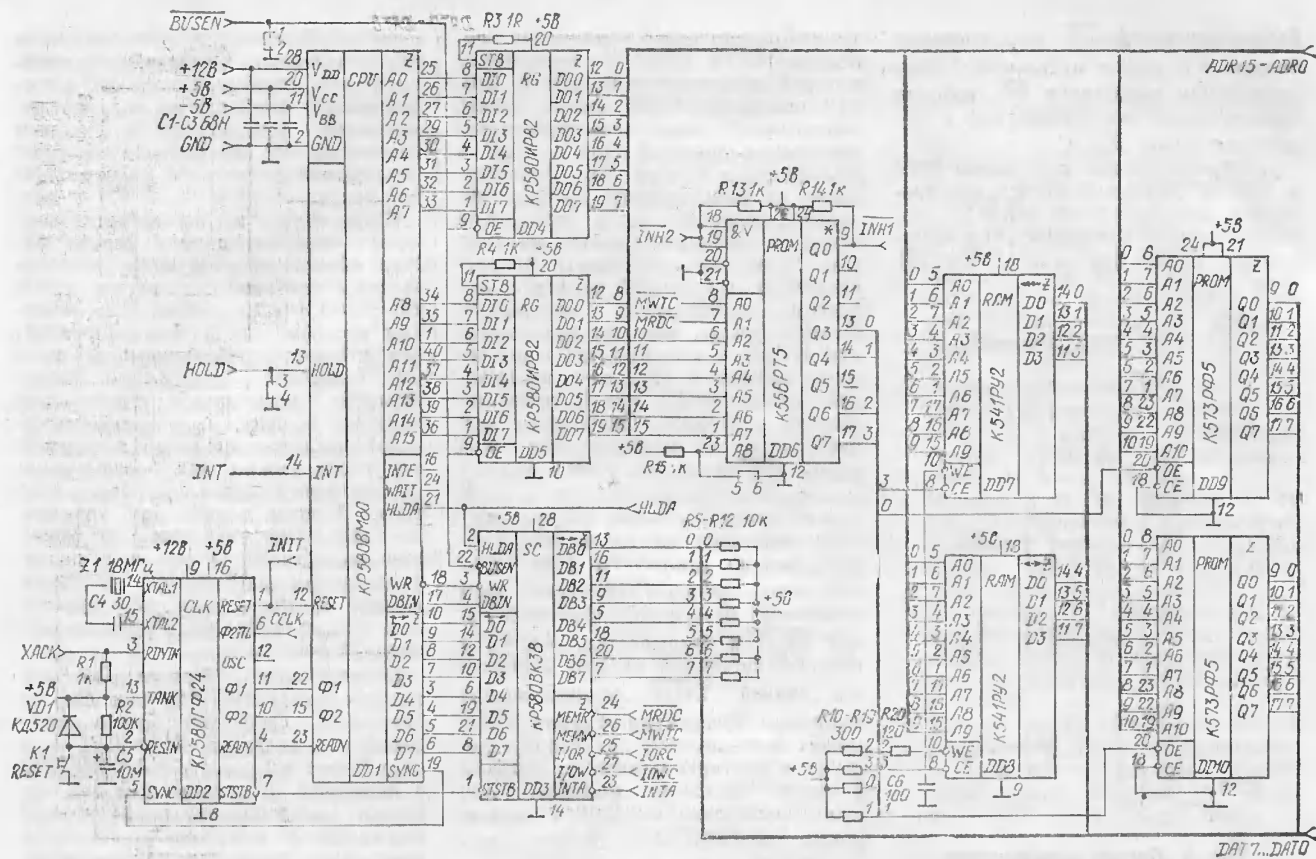


Рис. 2. Принципиальная схема микроконтроллера. Часть 2

более, поэтому на плате МК следует предусмотреть одно или два места для установки микросхем УФРПЗУ типа К573РФ2/РФ4/РФ5/РФ6. Могут быть использованы также любые другие ПЗУ объемом 2...32К байт, совместимые по разводке с вышеупомянутыми. Применение двух младших кристаллов К573РФ2/РФ5 данного семейства или одного кристалла К573РФ41 обеспечивает минимальный объем ПЗУ в 4К байт. Две 32К-байтные микросхемы покрывают все адресное пространство МК.

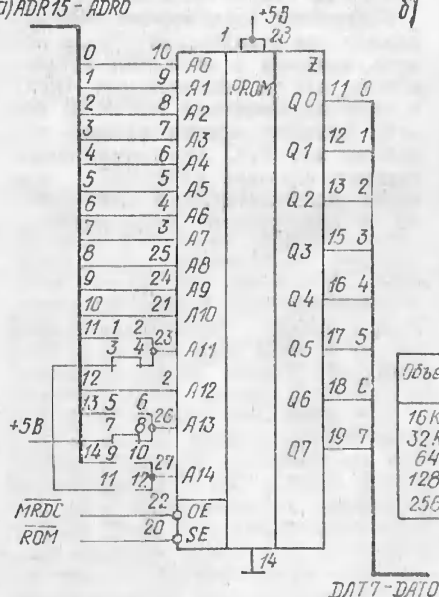
Рассматриваемый МК имеет два места для установки УФРПЗУ К573РФ2/РФ5. При включении питания или нажатии клавиши RESET управление передается на ячейку памяти с нулевым адресом, поэтому одна микросхема занимает первые 2К байт адресного пространства МК, вторая — область 0F800H...0FFFFH в старшей части пространства. По усмотрению пользователя вторая микросхема может быть перемещена в любую другую область.

Другой подход состоит в использовании универсальной панельки (рис. 4), допускающей применение микросхем с колодкой, аналогичной К573РФ2/РФ4/РФ5/РФ6 [1].

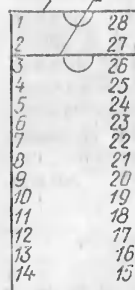
Оперативная память МК занимает объем в 1К байт и реализована на двух микросхемах ОЗУ статического типа К541PУ2 с организацией 1Кx4. Базовый адрес ОЗУ равен

1000H. Лучше всего для этой цели подходят микросхемы с байтовой организацией, например, 2К-байтная микросхема К537PУ8/PУ9. Эти микросхемы наряду с линиями вы-

а) ADR15 - ADRO



б)



| Объем | Тип УФРПЗУ | Перемычки | Способ установки |
|-------|-------------|-----------------|------------------|
| 16К | К573PФ2/РФ5 | 3-4, 7-8 | 2 |
| 32К | К573PФ4 | 1-2, 7-8 | 2 |
| 64К | К573PФ5 | 1-2, 7-8, 11-12 | 1 |
| 128К | 27128 | 1-2, 5-6, 11-12 | 1 |
| 256К | 27256 | 1-2, 5-6, 9-10 | 1 |

Рис. 4. Универсальная панелька: схема подключения (а), установка микросхемы (б)

борки кристалла CE и разрешения записи WE имеют отдельную линию разрешения считывания OE, которая подключается непосредственно к командной линии MRDC.

Для подключения кристаллов ОЗУ и ПЗУ к системной магистрали требуется дополнительная логика, которую удобно реализовать на биполярных ППЗУ [2] или ПЛИМ [3] (рис. 5).

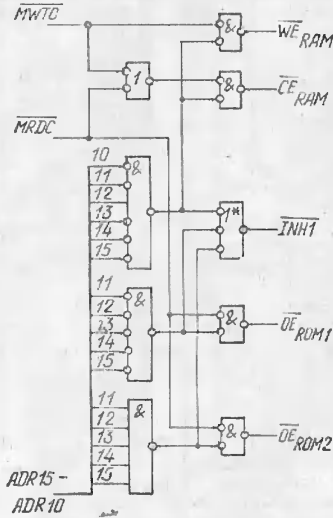


Рис. 5. Логика подключения приборов памяти

Память МК расширяется с помощью дополнительной платы памяти, достаточной для покрытия всего адресного пространства в любых сочетаниях «оперативная — постоянная память». Системный сигнал $\overline{INH1}$ является общим сигналом выборки ОЗУ/ПЗУ, размещенных на плате микроконтроллера. Он служит для запрета работы системной памяти в моменты обращения ЦП к местным ресурсам. Сигнал $\overline{INH1}$ формируется по схеме с открытым коллектором для организации «монтажно-го ИЛИ» с аналогичными сигналами запрета от других модулей системы. Для построения логики

(рис. 5) используется младшая половина ППЗУ К556РТ5, содержимое которой представлено на рис. 6. Вторая половина ППЗУ может быть запрограммирована для другого распределения областей ОЗУ и ПЗУ в пространстве памяти МК. Использование программируемой логики позволило сократить число корпусов на плате и стандартизировать монтажную схему вне зависимости от распределения физической памяти в адресном пространстве микроЭВМ. Следует учесть, что при использовании ППЗУ в моменты его переключения возможно появление на выходах микросхемы кратковременных ложных выбросов. Опасность эти выбросы представляют только для линии WE приборов ОЗУ. Для их сглаживания рекомендуется использовать дополнительную RC-цепочку.

Размещенную на плате МК память можно отключить, если активизировать линию $\overline{INH2}$. При этом становится безусловно пассивной линия $\overline{INH1}$ и открывается доступ к внешней системной памяти. Управление линией $\overline{INH2}$ осуществляется внешними средствами. Сигнал $\overline{INH2}$ может быть полезен при запрете доступа в местную память со стороны внешних активных модулей. Для этого достаточно выполнить соединение $\overline{INH2} = \text{BUSEN}$. Можно ввести другой тип запрета $\overline{INH3}$, который отключает память на плате, но не разрешает доступ к системной памяти ($\overline{INH3} = 0$). Для этого следует сигнал $\overline{INH3}$ завести на старшую адресную линию ППЗУ и использовать ее вторую половину. Запрет такого типа может быть полезен при отображении портов ввода-вывода на память системы.

Подсистема ввода-вывода МК содержит три стандартных канала обмена данными с внешними устройствами: два последовательных ИРПС и один параллельный ИРПР. В основу последовательных каналов положены две БИС программируемого связного адаптера КР580ВВ51, которые инициализируются для работы в асинхронном старт-стоповом

режиме. Один канал резервируется для подключения дисплея, выполняющего роль системной консоли, а второй может быть использован по усмотрению пользователя. Типовым вариантом его применения служит реализация радиальной межмашинной связи.

Встроенная в последовательные каналы связи выходная буферная логика обеспечивает передачу «Лог.0» высоким уровнем напряжения +12 В ($R_{out} = 1,3 \text{ кОм}$), а «Лог.1» — низким уровнем -12 В ($R_{out} = 300 \text{ Ом}$), что отвечает требованиям на стык С2 (стандарт RS-232-C). Предусмотрена возможность устранения инверсии во входных и выходных каналах последовательного интерфейса, что очень важно для организации межмашинной связи магистрального типа. В этом случае для управления локальной микросетью используется магистральная линия с последовательным протоколом передачи. Доступ к магистрали регулируется с помощью передачи специального маркера доступа.

Параллельный 8-разрядный интерфейс типа ИРПР реализован на базе БИС программируемого периферийного адаптера КР580ВВ55 и двух буферных регистров КР580ИР83 с инверсией. Максимальный ток нагрузки I_{out} выходной шины данных составляет 32 мА. Каналы РВ и РА программируются для работы в режиме соответственно ввода и вывода со стробированием. Две свободные линии РС4 и РС5 порта РС могут использоваться по усмотрению пользователя, например, для формирования выходных сигналов $\overline{INH2}$ и тестового контроля.

Средства поддержки реального времени. Микроконтроллеры обычно работают в режиме реального времени [4]. В микроконтроллере ПМС 1204 этот режим поддерживается системным таймером и контроллером обработки прерываний. Построенный на базе БИС КР580ВВ53 системный таймер содержит три 16-разрядных счетчика. Первый резервируется для формирования системных меток реального времени. Обычно они следуют с частотой

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| .000. | FF | F6 | FF | FF | FF | F6 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .010. | FF | 7E | 3E | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .020. | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .030. | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .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. 6. Карта прошивки ППЗУ логики выборки кристаллов памяти

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| .000. | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .010. | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .020. | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .030. | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF |
| .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. 7. Карта прошивки ППЗУ логики выборки периферийных кристаллов ввода-вывода

100 Гц, которая может быть изменена пользователем. Второй отведен для отработки временных пауз и интервалов в единицах системного времени и служит для индикации момента наступления следующего события. При частоте следования меток в 100 Гц максимальная длительность интервала составляет 655,36 с. На основе третьего построен генератор скорости передачи данных через последовательные каналы. Использование счетчиков в других целях не предусмотрено.

Восьмиуровневая система прерываний реализована на основе БИС КР580ВН59. Часть линий IR5...IR1 используется для приема запросов от внутрисплатных источников: счетчика меток реального времени (IR1), счетчика временных интервалов (IR2), выходного канала ИРПР (IR3), приемников ИРПР (IR4 и IR5). Для уменьшения времени отклика на запросы каждому источнику отведен индивидуальный уровень прерывания. Запрет отдельных прерываний осуществляется установкой соответствующих бит в регистре маски контроллера КР580ВН59.

Линии IR0, IR6, IR7 выведены на системную магистраль для приема запросов от внешних источников, число которых обычно не превышает трех. Для увеличения числа уровней прерывания, например, методом каскадирования необходимо на раземе системной магистрали предусмотреть три линии каскадного расширения CAS2...CAS0.

Массивы портов периферийных кристаллов размещены в пространстве ввода-вывода микросистемы начиная с адреса 12Н. Последовательность размещения кристаллов и их базовые адреса следующие: КР580ВН59 (12Н), КР580ВВ55 (14Н), первый КР580ВВ51 (18Н), второй КР580ВВ51 (1АН), КР580ВВ53 (1СН). Для более равномерного распределения нагрузки для адресации внутрикристалльных портов используются линии ADR8, ADR9 старшей половины адресной шины. Логика выборки кристаллов реализована на младшей половине ППЗУ К556РТ5, карта прошивки которой представлена на рис. 7. Заведенный на один из входов выборки кристалла ППЗУ К556РТ5 сигнал BUSEN обеспечивает выключение местных средств ввода-вывода, если магистраль захвачена внешним модулем. В противном случае возможно нарушение работы системы.

Ядро программного обеспечения. Строгая специализация аппаратных ресурсов МК обеспечивает их стандартную стыковку с программным обеспечением пользователя. Появляется возможность разработки ядра ПО — специальной программной компоненты общесистемного типа, образующей центральную часть приклад-

ных программ. Ядро реализует непосредственный стык с аппаратурой МК, освобождая программиста от утомительных процедур прямого управления его работой.

Применение микроконтроллера. Выбор наиболее часто используемых функций и их интеграции в полном объеме на одной плате обеспечили сочетание стандартизации аппаратуры МК с универсальностью применения. Микроконтроллер ММС 1204, дополненный платой системного ОЗУ и средствами связи с внешней памятью, может служить центральным элементом микроЭВМ. В автономном варианте МК может быть полезен при построении сетевого контроллера, через который осуществляется подключение к локальной сети дисплеев, принтеров, накопителей и других периферийных устройств со стандартными интерфейсами типа ИРПР или ИРПС.

В сочетании с модулями для программирования ПЗУ [5, 6] микроконтроллер образует автономный программирующий комплекс (рис. 8).

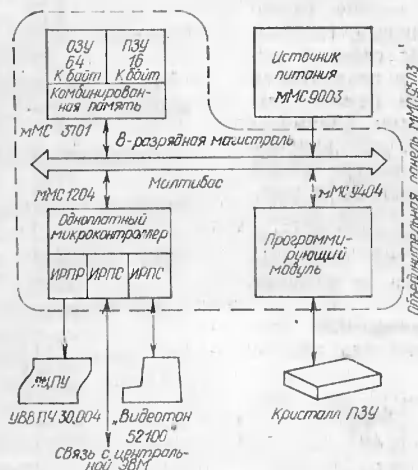


Рис. 8. Структурная схема программатора ММС 0602

Программное обеспечение такого комплекса требует объемов памяти, которые превышают ресурсы микроконтроллера, поэтому была разработана специальная плата ММС 3701 системной памяти комбинированного типа, содержащая до 64К байт ОЗУ и до 16К байт ПЗУ. В состав комплекса также входят объединительная панель, дисплей, принтер и источник питания ММС 9003.

Расширению области применения микроконтроллера способствует его совместимость с ПО серийно выпускаемого вычислительного комплекса СМ 1800. Для микроконтроллера этот комплекс может рассматриваться как инструментальная система проектирования программного обеспечения.

Справки по телефону: 408-62-44, г. Москва.

ЛИТЕРАТУРА

1. Щербаков О. А. Физические основы записи информации в ПЗУ // Микропроцессорные средства и системы.—1985.—№ 3.—С. 72—75.
2. Щелкунов Н. Н., Дианов А. П. ПЗУ вместо произвольной логики // Микропроцессорные средства и системы.—1986.—№ 1.—С. 83—86.
3. Щелкунов Н. Н., Дианов А. П. Процедуры программирования логических матриц // Микропроцессорные средства и системы.—1986.—№ 2.—С. 71—76.
4. Щелкунов Н. Н., Дианов А. П. Программирование микросистем реального времени // Микропроцессорные средства и системы.—1985.—№ 4.—С. 31—35.
5. Дианов А. П., Щелкунов Н. Н. Модуль программирования микросхем ПЗУ // Микропроцессорные средства и системы.—1985.—№ 3.—С. 80—83.
6. Дианов А. П., Щелкунов Н. Н. Технические средства программирования логических схем // Микропроцессорные средства и системы.—1986.—№ 2.—С. 77—80.

Статья поступила 26 марта 1986 г.

МИКРОПРОЦЕССОРЫ В ПОЛИТЕХНИЧЕСКОМ

В 1987 году журнал проведет в Политехническом музее учебный цикл семинаров, ориентированный на учителей и методистов по школьной информатике:

Учителю о компьютере: опыт преподавания

Школьная информатика: предмет, методы и стиль преподавания
 Вводный курс компьютерной грамотности для педагога
 Типы школьных ЭВМ и программное обеспечение кабинета информатики
 Организация школьного практикума по программированию
 ЭВМ в школе — обучение игрой: место ЭВМ в преподавании традиционных школьных дисциплин
 ЭВМ и учитель: итоги первых двух лет преподавания школьной информатики

Справки о семинарах можно получить по телефону: 923-00-19.

Л. Н. Преснухин, А. Н. Соловьев, Н. П. Кузнецов,
О. Л. Семичастнов, К. П. Разумовский

КОНТРОЛЛЕРЫ ИНДИКАЦИИ И КЛАВИАТУРЫ НА ОСНОВЕ НЕСКОММУТИРОВАННЫХ ВЕНТИЛЬНЫХ МАТРИЦ

В соответствии с принятой в микропроцессорной технике модульной архитектурой систем [1, 2] устройство операторного ввода-вывода информации (УОВВ) выполняется в виде структурно завершенного узла. В его состав входят контроллеры клавиатуры и устройств индикации, обеспечивающие сопряжение с системным интерфейсом. Кроме контроллеров, УОВВ содержит узлы логического и электрического сопряжения с клавиатурой и индикаторным табло.

Предлагаемый функциональный комплект включает устройства отображения информации и клавишного ввода, датчик системного времени и программируемый таймер. Комплект ориентирован на использование широкой номенклатуры индикаторов и клавиатур. Обеспечивает совместимость с матричными (типа АЛ340) и сегментными (типа АЛ304, АЛ318) индикаторами, имеющими различную технологическую основу (полупроводниковые, вакуумные люминесцентные, газоразрядные), с встроенными (в индикаторы) схемами управления и без них. При этом аппаратно реализуются хранение кодов символов во внутренней памяти устройства индикации и управление режимом динамической индикации с программно заданной скважностью (яркостью) подсвета. Также аппаратно внутри УОВВ реализуются обработка сигналов с клавиатуры и генерация соответствующих им стандартных кодов КОИ-7 (ГОСТ 13052—74). Благодаря организации перечисленных автономных режимов работы УОВВ центральный процессор микроЭВМ освобождается от решения задач ввода-вывода.

Связь УОВВ с другими модулями микроЭВМ осуществляется через стандартную магистраль МПП,

причем возможно подключение аппаратуры УОВВ к микропроцессорам К1801ВМ1, К1801ВМ2 [3] и вычислительным устройствам на их основе без промежуточных сопрягающих схем.

В состав комплекта входят три БИС: контроллер устройств индикации БИС КУИ—К1806ВП1-092; контроллер клавиатуры (БИС КК) и программируемый таймер—К1806ВП1-093; знакогенератор на основе БИС ПЗУ (типа К556РТ7, К596РЕ1 и т. п.). Программной поддержкой служат драйверы управления индикацией и клавиатурой (объектный перемещаемый модуль) и тестовые программы проверки функционирования (объектный перемещаемый модуль). Программные драйверы и тестовые программы комплекта написаны в системе команд микроЭВМ «Электроника 60М», что позволяет применять их совместно со стандартным программным обеспечением. Комплект сопровождается инструкцией по применению, которая содержит описание и расчет основных вариантов реализации УОВВ.

БИС КУИ и БИС КК выполнены по КМОП-технологии и размещены в 42-выводных корпусах (рис. 1, 2). Для их работы необходимо напряжение питания 5 В, потребляемая мощность не превышает 10 мВт.

| | | | | |
|----|-------|-----|-----------------|----|
| 1 | D5 | DDD | V _{cc} | 42 |
| 2 | D6 | | D4 | 41 |
| 3 | D7 | | D3 | 40 |
| 4 | WE | | D2 | 39 |
| 5 | CLC | | D1 | 38 |
| 6 | TA | | DO | 37 |
| 7 | TB | | DOU7 | 36 |
| 8 | TC | | AD0 | 35 |
| 9 | SYNCO | | AD1 | 34 |
| 10 | CLC | | AD2 | 33 |
| 11 | BL | | AD3 | 32 |
| 12 | SA | | AD4 | 31 |
| 13 | SB | | AD5 | 30 |
| 14 | SC | | AD6 | 29 |
| 15 | K0 | | AD7 | 28 |
| 16 | K1 | | CS | 27 |
| 17 | K2 | | SYNC | 26 |
| 18 | K3 | | DIN | 25 |
| 19 | K4 | | BS | 24 |
| 20 | CM | | RPLY | 23 |
| 21 | GND | | BKL | 22 |

Рис. 1. Условное графическое изображение БИС К1806ВП1-092

| | | | | |
|----|-------|-----|-----------------|----|
| 1 | Y3 | CTC | V _{cc} | 42 |
| 2 | Y4 | | Y2 | 41 |
| 3 | Y5 | | Y1 | 40 |
| 4 | Y6 | | ADD | 39 |
| 5 | Y7 | | AD7 | 38 |
| 6 | X9 | | AD2 | 37 |
| 7 | X8 | | AD3 | 36 |
| 8 | X7 | | AD4 | 35 |
| 9 | X6 | | AD5 | 34 |
| 10 | X5 | | AD6 | 33 |
| 11 | X4 | | AD1 | 32 |
| 12 | X3 | | EC2 | 31 |
| 13 | X2 | | CS | 30 |
| 14 | X1 | | SYNC | 29 |
| 15 | X0 | | RPLY | 28 |
| 16 | SHIFT | | DOU7 | 27 |
| 17 | CLC | | DIN | 26 |
| 18 | CTRL | | IAKI | 25 |
| 19 | INIT | | IAKO | 24 |
| 20 | RP1 | | VIRQ | 23 |
| 21 | GND | | EVNT | 22 |

Рис. 2. Условное графическое изображение БИС К1806ВП1-093

Благодаря этому комплект обеспечивает построение экономичных УОВВ и создает возможность для их комплексной микроминиатюризации. Назначение выводов этих БИС показано в табл. 1, 2. Основные технические характеристики УОВВ на базе комплекта приведены ниже.

| | | |
|---|-------|---|
| Число знакомест, не менее | | N1×12, где N1—число БИС КУИ |
| Скважность подсвета | | 6...90 |
| Частота смены кадра на дисплее, Гц | | 400, при тактовой частоте 1152 кГц |
| Число клавиш на клавиатуре, не менее | | N2×80, где N2—число БИС КК в составе УОВВ |
| Частота сигналов программируемого таймера, Гц | | 2...400 |
| Диапазон рабочих температур °С, не менее | | ±50 |

Наличие в комплекте специализированных БИС КУИ и БИС КК позволяет существенно сократить число элементов УОВВ и повысить его надежность и эксплуатационные характеристики.

Назначение выводов БИС К1806ВП1-092

| Вывод | Назначение |
|---------|--|
| 1...3 | Информационные выходы D5, D6, D7 |
| 4 | Выход управления внешним регистром WE |
| 5 | Выход тактовой частоты CLC |
| 6...8 | Выходы сигналов сдвига TA, TB, TC |
| 9 | Вход-выход сигнала синхронизации SYNCO |
| 10 | Вход (технологической) тактовой частоты CLCT |
| 11 | Выход сигнала подсвета BL |
| 12...14 | Выходы сигналов управления питанием SA, SB, SC |
| 15...19 | Выходы сигналов выбора столбцов K0, K1, K2, K3, K4 |
| 20 | Вход сигнала выбора режима работы CM |
| 21 | Общий вывод GND |
| 22 | Выход сигнала выбора внешнего устройства BKL |
| 23 | Выход сигнала ответа приемника информации RPLY |
| 24 | Вход сигнала выбора адреса BS |
| 25 | Вход сигнала управления вводом данных DIN |
| 26 | Вход сигнала синхронизации обмена SYNC |
| 27 | Вход сигнала выбора устройства CS |
| 28...35 | Входы-выходы магистрали адреса — данных AD7...AD0 |
| 36 | Вход сигнала управления выводом данных DOUT |
| 37...41 | Информационные выходы D0...D4 |
| 42 | Напряжение источника питания U _{cc} +5В |

Назначение выводов БИС К1806ВП1-093

| Вывод | Назначение |
|---------|--|
| 1...5 | Входы-выходы линий Y5, Y4, Y3, Y2, Y1 |
| 6...15 | Входы линий X9, X8, X7, X6, X5, X4, X3, X2, X1, X0 |
| 16 | Вход сигнала переключения регистров SHIFT |
| 17 | Вход тактовой частоты CLC |
| 18 | Вход сигнала модификации кодов CTRL |
| 19 | Вход сигнала начальной установки INIT |
| 20 | Вывод для подключения времязадающей цепочки RPI |
| 21 | Общий вывод GND |
| 22 | Выход сигнала таймера EVNT |
| 23 | Выход сигнала адресного прерывания VIRQ |
| 24 | Выход сигнала разрешения прерывания IAKO |
| 25 | Вход сигнала разрешения прерывания IAKI |
| 26 | Вход сигнала управления вводом данных DIN |
| 27 | Вход сигнала управления выводом данных DOUT |
| 28 | Выход сигнала ответа приемника информации RPLY |
| 29 | Вход сигнала синхронизации обмена SYNC |
| 30 | Вход сигнала выбора устройства CS |
| 31 | Вход сигнала модификации адреса прерывания EC2 |
| 32...39 | Входы-выходы магистрали адреса данных AD1, AD6, AD5, AD4, AD3, AD2, AD7, AD0 |
| 40, 41 | Входы-выходы линий Y7, Y6 |
| 42 | Напряжение источника питания U _{cc} +5В |

Контроллер устройств индикации

БИС КУИ ориентирована на работу с дисплеями на основе дискретных индикаторов. В состав БИС входят следующие функциональные узлы (рис. 3):

двухнаправленный трехстабильный буфер для подключения к магистрали адреса-данных микроЭВМ;

блок синхронизации для управления обменом информацией через МПИ;

компаратор адресов, используемый при адресном обращении к регистрам БИС КУИ. При дешифрации адреса используются сигналы с линий AD0...AD7 и CS, при этом значение сигнала на линии AD7 анализируется в зависимости от уровня сигнала на входе BS. Сигнал BKL используется для обращения к регистрам БИС КК;

ОЗУ для приема, хранения и выдачи 8-разрядных кодов 12-ти символов;

блок управления записью, формирующий сигналы управления работой параллельно-последовательных внешних регистров. Сигнал WE определяет режим работы регистра, а сигналы TA, TB, TC сопровождают запись данных в последовательный регистр с частотой сигнала на входе CLC;

блок формирования диаграммы подсвета для управления отображением информации на дисплее, формирующий сигнал подсвета BL, сигналы управления включением питания групп знакомест SA, SB, SC, сигналы K0...K4, с помощью которых можно управлять отображением информации на матричных индикаторах. Сигнал CM модифицирует временную диаграмму под конкретный тип используемых индикаторов (матричные или сегментные). Вход-выход SYNCO позволяет синхронизировать работу нескольких БИС КУИ;

выходной мультиплексор, осуществляющий выдачу на выходы D0...D7 кодов символов из регистров ОЗУ в последовательности, определяемой порядковым номером регистра. Кроме того, на выходе D0 происходит побитовая выдача информации из каждого регистра ОЗУ с частотой выдачи, соответствующей тактовой частоте на входе CLC;

регистры режима и состояния. Семиразрядный регистр режима доступен по записи, значения его разрядов определяют режим работы БИС КУИ. Трехразрядный регистр состояния содержит информацию о том, какие из регистров ОЗУ опрошены, доступен по чтению.

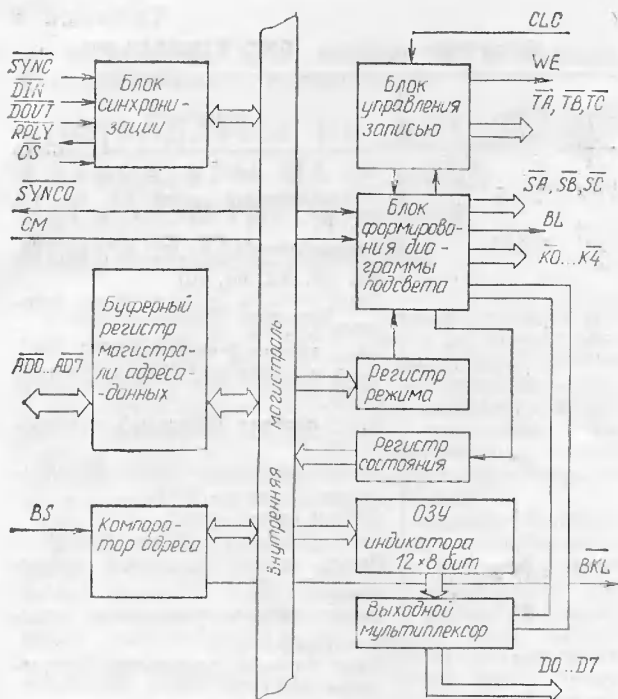


Рис. 3. Структурная схема контроллера устройств индикации (БИС К1806ВПИ-092)

Рассмотрим подробнее форматы двух последних регистров. Разряды 0, 1, 2 регистра режима содержат код длительности подсвета данных на дисплее. В зависимости от этого кода скважность сигнала подсвета BL принимает значения из ряда: 6; 7,5; 10; 15; 22,5; 30; 45; 90. При «1» в третьем разряде устанавливается технологический режим работы БИС, при «0» — рабочий. Биты в 4, 5, 6-м разрядах разрешают появление сигналов включения питания SA (знакоместа 1..4), SB (знакоместа 5..8) и SC (знакоместа 9..12) соответственно.

Применение БИС контроллера устройств индикации

БИС КУИ может использоваться как с 8-сегментными индикаторами (типа АЛС321, АЛС348 и подобных), так и с матричными индикаторами, имеющими встроенную схему управления [4].

В первом случае устройство индикации содержит (рис. 4), помимо БИС КУИ, двенадцать индикаторов, дешифратор номера знакоместа и схемы сопряжения. Каждое знакоместо представляет собой набор из восьми сегментов, которые имеют общее управление сигналами с выходов DO...D7 БИС КУИ. Выбор соответствующего знакоместа происходит с помощью дешифратора, реализованного на трех микросхемах К564ЛН1. Комбинация сигналов на выходах КО...К3 и SA, SB, SC определяет выбор одного из двенадцати знакомест, при этом к выходам DO...D7 подключается соответствующий регистр ОЗУ. Сигнал с выхода BL регулирует яркость свечения индикаторов благодаря изменению времени подсвета. Режим работы БИС КУИ с индикаторами первого типа задается подачей на вход CM высокого уровня сигнала.

Более широкими возможностями по отображению информации обладают матричные индикаторы, в которых отдельное знакоместо представляет собой матрицу с организацией 5x7 точек [5]. Устройство индикации в этом случае включает функциональные элементы (рис. 5):

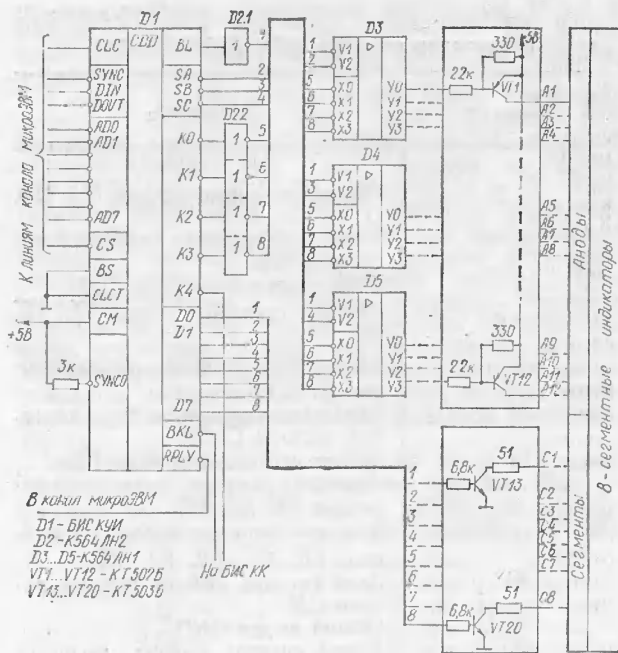


Рис. 4. Функциональная схема устройства индикации на основе 8-сегментных индикаторов

БИС КУИ для сопряжения с магистралью микроЭВМ и управления отображением информации; генератор символов для преобразования кодов КОИ-7 в соответствии с системой кодирования для индикаторного поля 5x7 точек. Он включает ПЗУ символов (микросхема К556РТ7 или аналогичная) и параллельно-последовательный регистр-преобразователь (микросхема К564ИР6);

схемы электрического согласования для управления подачей питания и столбцами индикаторов;

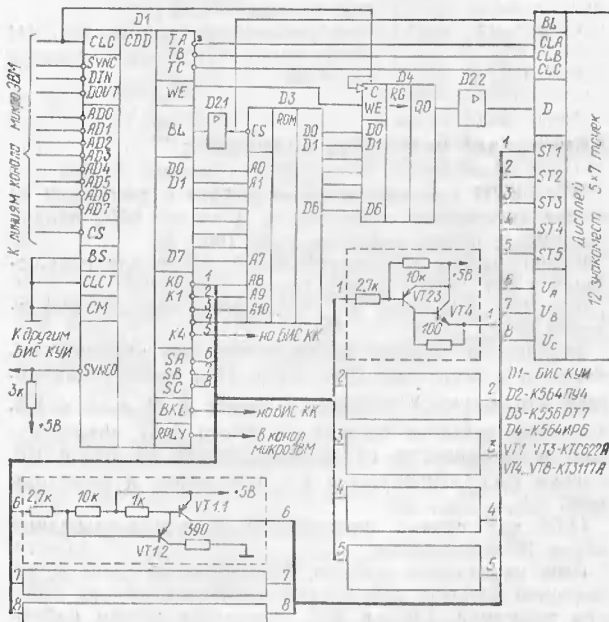


Рис. 5. Функциональная схема устройства индикации на основе матричных индикаторов со встроенными схемами управления

три четырехзнакоместных индикатора с встроенными схемами управления.

Запись информации в ОЗУ БИС КУИ производится с МПИ в соответствии с процедурами режима Вывод [3]. Выводы БИС КУИ могут быть подключены без дополнительного согласования к линиям магистрали микропроцессоров К1801ВМ1, К1801ВМ2. Отображение записанной в ОЗУ информации осуществляется независимо от процессора с помощью аппаратуры БИС КУИ. При подаче на вход CLC тактовой частоты 1152 кГц микросхема обеспечивает режим динамической индикации с частотой 400 Гц.

В начале каждого цикла сканирования информация записывается во внутренние регистры индикаторов по сигналам с выходов ТА, ТВ, ТС. При этом сначала заносится информация для первых столбцов всех знакомест. На адресные входы ПЗУ символов при этом поступает код отображаемого символа и код номера столбца (с выходов К0, К1, К2). После занесения данных сигналом ВЛ разрешается подсвет этих данных в первых столбцах, так как формируется сигнал К0, который через транзисторный ключ подается на входы ST1 индикаторов. Затем аналогично осуществляется отображение информации во вторых столбцах всех знакомест. После индикации данных пятого столбца цикл сканирования заканчивается.

Для регулировки яркости в зависимости от условий внешней освещенности предусмотрено программное изменение скважности сигнала ВЛ (в регистр режима заносится соответствующий код). Индикаторы включаются только на время занесения данных и действия сигнала ВЛ, что значительно снижает потребляемую мощность. В остальное время ключевые схемы отключают индикаторы от источника питания. При использовании нескольких БИС КУИ для управления работой дисплея, имеющего более 12 знакомест, можно синхронизировать их работу, объединив выходы SYNCO.

Контроллер клавиатуры

Организация диалога оператора и микроЭВМ предполагает наличие клавиатуры и таймера. В БИС КК реализована электронная часть клавиатуры и таймера (рис. 6):

двухнаправленный трехстабильный буфер для подключения к магистрали адреса-данных микроЭВМ;

блок синхронизации для управления обменом информацией через МПИ. В отличие от аналогичного блока БИС КУИ здесь имеется возможность организации обмена данными как под управлением программы, так и в режиме прерывания (выводы VIRQ, IAKO, IAKI);

схема формирования адреса вектора прерывания, которая по запросу прерывания от таймера формирует адрес 104_8 , а при вводе данных с клавиатуры формирует адрес 60_8 (при $EC2 = \langle 1 \rangle$) или 274_8 (при $EC2 = \langle 0 \rangle$);

кодеры X, Y для формирования семибитных стандартных кодов символов в соответствии с сигналами, поступающими от клавиатуры по линиям X0...X9 и Y1...Y7. Посредством управляющих сигналов CTRL, SHIFT — можно модифицировать таблицу кодирования; регистры данных и таймера, имеющие на системной магистрали общий адрес. Регистр данных доступен с МПИ по чтению; в него заносится код из кодеров и хранится до момента чтения. В регистр таймера записывается код коэффициента пересчета программируемого счетчика;

5-разрядный счетчик, коэффициент пересчета которого изменяется в соответствии с числом в регистре таймера. На его вход поступают счетные импульсы либо с частотой сигнала на входе CLC, либо с частотой, в восемь раз меньшей;

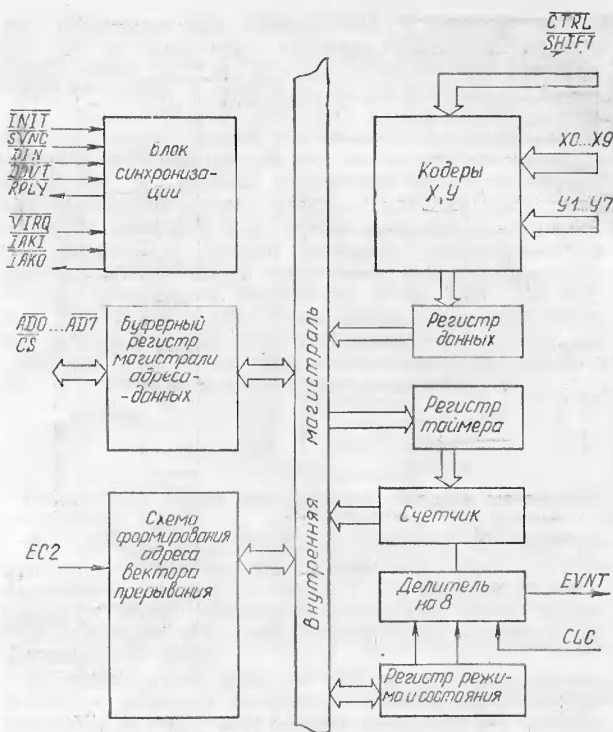


Рис. 6. Структурная схема контроллера клавиатуры (БИС К1806ВPI-093)

делитель частоты на восемь для управления входом счетчика. Он имеет выход EVNT, сигнал с которого может использоваться для подсчета системного времени;

регистр режима — состояния для управления работой устройства. В его разрядах имеются биты для маскирования сигнала на выходе EVNT, запросов прерываний от таймера и клавиатуры. Два разряда содержат бит готовности клавиатуры и бит управления входом счетчика.

Реализованная в БИС КК логика прерываний дает таймеру более высокий приоритет по отношению к клавиатуре. Кодеры X, Y организованы таким образом, что обеспечивается защита от влияния дребезга контактов и одновременного нажатия нескольких клавиш.

Устройство ввода данных с клавиатуры

В состав устройства (рис. 7) входят клавиатура, БИС КК, три переключателя (SA1, SA2, SA3) и элементы электрического сопряжения. При наличии в УОВВ одной БИС КК клавиатура может содержать до 80 кнопок, организованных в матрицу 8×10 . Каждая кнопка содержит пару нормально разомкнутых контактов. При замыкании одной из пар контактов замыкаются соответствующие шины X и Y; при этом образуется делитель напряжения на резисторах, подключенных к этим шинам. Для исключения ложного срабатывания схемы кодеров сигнал запуска сглаживается RC-цепью, подключенной к выводу RPI.

Переключатели SA1, SA2, SA3 позволяют модифицировать таблицу кодирования сигналов, поступающих с клавиатуры. Все формируемые коды соответствуют кодам КОИ-7 ГОСТ 13052-74 [6]. На выходе EVNT формируется меандр с частотой 50 Гц, который может использоваться для подсчета системного времени.

Для сокращения сроков разработки и внедрения УОВВ на основе данного комплекта были созданы программы, адаптированные к вычислительным устройствам с системой команд микроЭВМ «Электроника

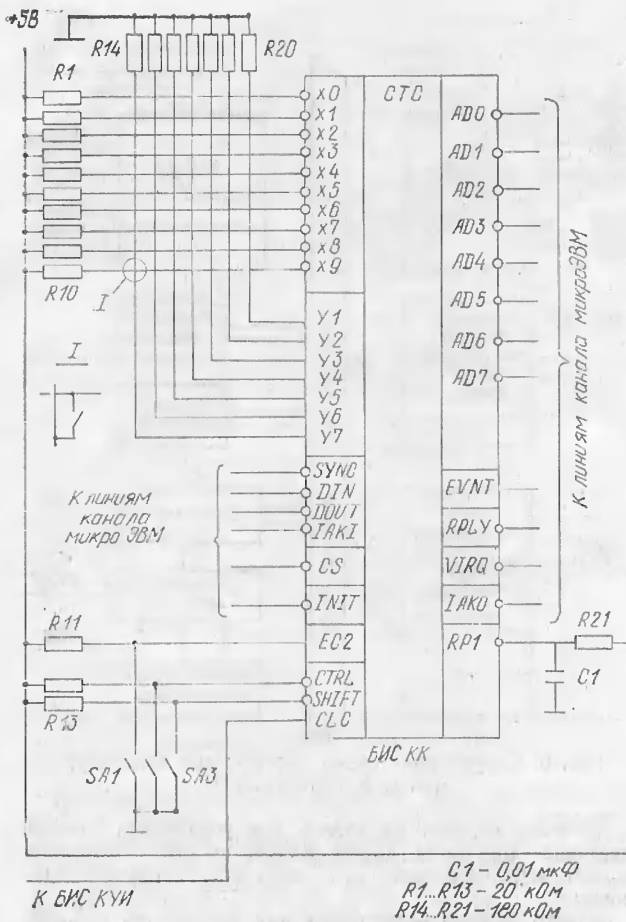


Рис. 7. Функциональная схема устройства ввода данных с клавиатуры

60М». ПО комплекта включает драйвер пульта оператора (объектный перемещаемый модуль); программы — утилиты, ориентированные на специальные применения УОВВ и обеспечивающие оптимизацию тактико-технических характеристик микроЭВМ (объектный перемещаемый модуль); тестовые программы проверки работоспособности УОВВ на этапах отладки и эксплуатации (объектный перемещаемый модуль).

Драйвер пульта оператора включает в себя подсистему индикации и клавиатурного ввода. Подсистема индикации содержит драйвер индикатора, процедуры распределения ресурсов, сервисные процедуры.

Драйвер индикатора обеспечивает физический вывод на индикатор содержимого буфера индикации, а также управление яркостью свечения индикатора. Кроме того, он производит оптимизацию энергопотребления индикатора в зависимости от состава отображаемой информации. Так, в случае индикации пробельной строки, драйвер гасит индикатор. Процедуры распределения ресурсов индикатора определяют monopolное использование индикатора системой, задержек и одновременный вывод в буфер индикатора системной и прикладной программ с условным выводом на индикатор, т. е. виртуальный вывод информации. Сервисные процедуры подсистемы индикации не связаны с реальным выводом на индикатор; они осуществляют передачу данных между областями памяти.

Драйвер клавиатуры реализует запросы системных и прикладных задач на ввод информации с клавиатуры, обрабатывает ситуации ошибочного нажа-

тия клавиш и осуществляет передачу управления при нажатии специальных клавиш.

Тестовые программы (пользовательский и диагностический тесты) позволяют осуществлять оперативный контроль работоспособности УОВВ, формируют сообщение об ошибке и локализуют неисправность устройства с точностью до клавиши.

Работоспособность индикатора контролируется в результате отображения специальных графических символов.

ПО подготовлено к реализации в виде СВИС для использования в составе кремниевой операционной системы. Кроме того, ПО может быть подготовлено на любом из стандартных носителей — перфоленте, магнитной ленте, ГМД и т. д.

По вопросам приобретения комплекта можно обратиться по адресу: 103498, Москва, МИЭТ.

ЛИТЕРАТУРА

1. Нестеров П. В. Микропроцессоры. Архитектура и ее оценка. / Под ред. Л. Н. Преснухина. — М.: Высшая школа, 1984.
2. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы. — М.: Радио и связь, 1981.
3. Однокристальные микропроцессоры комплекта БИС серии К1801/В Л. Дыхуяня, Ю. И. Борщенко, В. Р. Науменков и др. // Микропроцессорные средства и системы. — 1984. — № 4. — С. 12—18.
4. Абрамов В. С., Леонов В. В., Платонов М. Н., Скарин В. К. / Многоэлементные матричные полупроводниковые индикаторы со схемами управления // Электронная промышленность. — 1982. — № 5—6. — С. 58—61.
5. Применение оптоэлектронных приборов: Пер. с англ. / Под ред. Ю. Р. Носова. — М.: Радио и связь, 1981.
6. ГОСТ 13052—74. Коды семибитные для обмена информацией. — М.: Изд-во стандартов, 1974.

Статья поступила 24 февраля 1986 г.

УДК 681.324

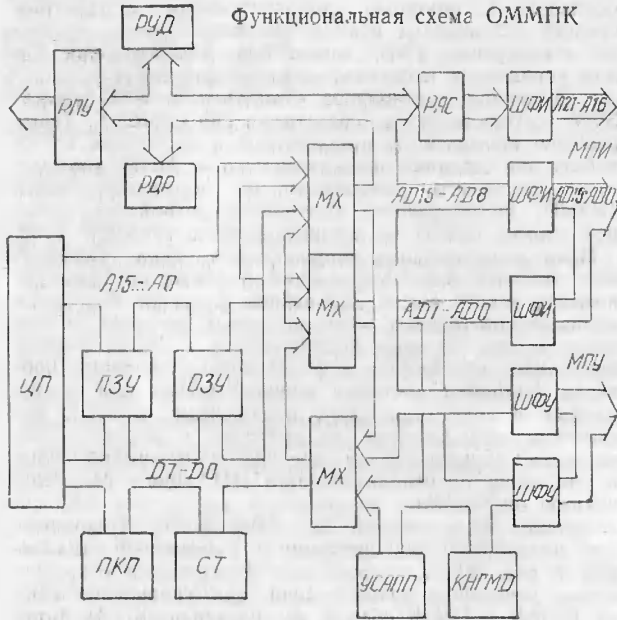
А. А. Шевченко

ОДНОПЛАТНЫЙ МНОГОЦЕЛЕВОЙ МИКРОКОНТРОЛЛЕР НА БАЗЕ БИС СЕРИИ КР580

Возможность совместного применения разнотипных микропроцессорных средств вычислительной техники (МСВТ) для построения гибких и высокоэффективных вычислительных систем имеет большое значение. В первую очередь это относится к наиболее доступным и пользующимся популярностью МСВТ «Электроника 60» и МСВТ, построенным на базе БИС серии КР580.

Развиваемый нами подход к решению задач управления устройствами ввода-вывода микроЭВМ «Электроника 60», ориентированными на побайтовый обмен, основан на использовании построенного на основе БИС КР580ИК80А сопроцессора ввода-вывода, что открывает возможность использования в составе МСВТ ряда «Электроника 60» не только аппаратных средств, но и программного обеспечения (ПО), разработанного для БИС серии К580. Функциональная схема такого сопроцессора — одноплатного многоцелевого микроконтроллера (ОММК), обеспечивающего возможность управления алфавитно-цифровым (или графическим) дисплеем (АЦД), печатающим устройством (ПУ) и накопителем на гибком магнитном диске (НГМД), показана на рисунке. Ядро ОММК образовано БИС центрального процессора (ЦП) и подключенными непосредственно к шинам ЦП БИС ПЗУ, ОЗУ, програм-

Функциональная схема ОММПК



мируемого счетчика-таймера (СТ) и программируемого контроллера прерываний (ПКП). С помощью сверхвысокочастотных мультиплексоров (МХ) с двух входов на один выход с тремя состояниями осуществляется мультиплексирование и усиление шин адреса и данных ЦП. К усвоенной мультиплексированной шине адреса/данных AD15...AD0 подключены: универсальный синхронный/асинхронный приемопередатчик (УСАПП), контроллер НГМД (КНГМД) — к шинам AD7...AD1, а также регистр вывода данных на печатающее устройство (РПУ), регистр управления дисковыми (РУД), регистр адреса сегмента (РАС) и регистр режима доступа к ресурсам (РДР) — к шинам AD15...AD8. Такое распределение нагрузки на внутренних шинах ОММПК позволяет обойтись без использования дополнительных буферных каскадов. Подключение ОММПК к межмодульному параллельному интерфейсу (МПИ) по ОСТ 11.305.903-80 с расширениями, принятыми в микроЭВМ «Электроника МС 1211», осуществляется через двунаправленные шинные формирователи (ШФИ). Шинные формирователи (ШФУ) обеспечивают реализацию дополнительной магистрали периферийных устройств (МПУ), также совместимой с требованиями ОСТ 11.305.903-80. Создание такой магистрали позволяет расширять состав подключенных к ОММПК устройств ввода-вывода (УВВ), требуя минимальных дополнительных технических средств.

На основе УСАПП в ОММПК реализуются широко распространенные стандартные последовательные интерфейсы ИРПС и стык С2, позволяющие подключить к ОММПК АЦГД. Обмен данными с УСАПП может осуществляться ЦП в режиме опроса и по прерываниям. Выбор режима и типа последовательного интерфейса — ИРПС или стык С2 — определяется коммутацией переключателя. Параллельный интерфейс ИРПР, реализованный на основе 8-разрядного буферного регистра, триггера и нескольких логических элементов серии К155, служит для подключения ПУ типа ДЗМ-180. Синхронизация вывода на печать обеспечивается прерываниями. Микроконтроллер выполняет передачу данных от подключенных к нему устройств на НГМД и обратно с промежуточной буферизацией в ОЗУ, а обмен данными с КНГМД осуществляется в реальном масштабе времени программами чтения и записи физического сектора на диске. Синхронизация обеспечивается задержкой сигнала подтверждения готовности

при обращении ЦП КР5801К80А к регистру данных КНГМД до появления сигнала требования данных (DRQ) БИС К1818ВГ93. Пересылка одного байта из ОЗУ ОММПК в КНГМД выполняется за 13,2 мкс (при тактовой частоте БИС ЦП равной 2,5 МГц). Выбранный способ обмена данными обеспечивает работу ОММПК с НГМД на 133 мм и на 203 мм дисках с одинарной и двойной плотностью записи, поскольку для физической записи одного байта на диск отводится не менее 16 мкс. Обмен данными завершается по прерыванию, запрос которого вырабатывается КНГМД по окончании выполнения заданной команды. Размер физического сектора на диске (128, 256, 512 или 1024 байт) задается при программировании режима работы КНГМД. Фрагмент программы записи физического сектора на диск, соответствующий записи в КНГМД 1 байта данных, приводится ниже,

| | | |
|--------|------|-------|
| WBYTE: | MOV | A M |
| | INX | H |
| | OUT | DISK |
| | JUMP | WBYTE |

Программа чтения физического сектора построена аналогично.

Доступом к вычислительным ресурсам ОММПК управляет устройство управления ресурсами (УУР). Анализируя сигналы с выхода РДР, старшие разряды адреса и слово состояния (СС) ЦП КР5801К80А, УУР разделяют адресное пространство ОММПК на две основные части:

1. *Область стека*, доступ к которой разрешается специальным разрядом обращения к МПИ при стековых операциях в РДР. Эта область содержит все подключенные к МПИ аппаратные средства микроЭВМ «Электроника 60». При обращении к МПИ 16-разрядный адрес БИС КР5801К80А дополняется до принятых в микроЭВМ «Электроника МС 1211» и «Электроника МС 1212» двадцати двух разрядов шестью разрядами РАС. Такое решение позволяет с помощью размещенной в ПЗУ подпрограммы обмена данными между ОЗУ ОММПК и системным ОЗУ осуществлять доступ к данным, хранящимся в любой области адресного пространства (4М байт) микроЭВМ «Электроника МС 1211» или «Электроника МС 1212».

2. *Область собственных ресурсов* (64 К байт), в которой, в свою очередь, выделяются ПЗУ, ОЗУ, область УВВ ОММПК и области подключенных к МПУ внешних ОЗУ и УВВ.

Устройство управления ресурсами выполнено по схеме шифратор — регистр кода операции обмена — дешифратор. Необходимость хранения кода операции в регистре обусловлена тем, что СС выдается БИС КР5801К80А по мультиплексированной во времени шине состояния/данных. Путем изменения кодировок установленного на контактной колодке ПЗУ К556РТ4 шифратора и коммутации переключек обеспечивается реконфигурация ресурсов ОММПК: перемещение начальных адресов ОЗУ и ПЗУ в поле адресации ЦП, отображение отведенной для МПУ области адресов на МПИ, выбор режима использования МПУ (8- или 16-разрядная шина с адресацией до 64К байт) и т. д. В момент записи данных в ПЗУ таймер формирует задержку сигнала готовности БИС КР5801К80А на 25 мс, что позволяет организовать программирование одной из двух БИС ПЗУ К573РФ2 ОММПК с помощью специальной программы перезаписи содержимого ПЗУ, хранящейся в ОЗУ, или части ПЗУ, не подлежащей перепрограммированию. Такая перезапись осуществляется по специальной команде оператора; при этом данные могут поступать от любого источника, назначенного в качестве логического устройства ввода.

Микроконтроллер обслуживает четыре уровня запросов прерывания от УВВ, подключенных к МПУ (средства обработки прерываний от устройств, подключенных к МПИ, не предусмотрены). При включении ОММПК в состав микроЭВМ «Электроника 60» такие

УВВ остаются в полном распоряжении 16-разрядного ЦП микроЭВМ. Это позволяет избежать проблем, связанных с обработкой прерываний и распределением ресурсов ввода-вывода между процессорами в многопроцессорной системе. Внутренние ресурсы ОММПК недоступны ЦП «Электроника 60». Обмен данными между ОММПК и ЦП 16-разрядной микроЭВМ происходит через буферы данных системного ОЗУ. Для выполнения операции ввода-вывода ЦП готовит буфер данных и загружает в выделенную в качестве «почтового ящика» область системного ОЗУ сообщение о необходимости выполнения требуемой операции, проверяет наличие запросов ввода-вывода, ожидающих обслуживания в почтовом ящике, выполняет требуемую операцию и сообщает о ее завершении. Проверка содержимого почтовых ящиков выполняется по прерываниям от программируемого счетчика — таймера. Аналогичная операция осуществляется по инициативе ОММПК при передаче данных, поступивших от АЦГД по последовательному каналу в системное ОЗУ.

Доступ ОММПК к системным ресурсам, подключенным к МПИ, выполняется в режиме прямого доступа к памяти (ПДП). Запрос ПДП устанавливается логическими цепями интерфейса в начале каждого машинного цикла БИС КР580ИК80А в результате декодирования адреса выбранных ресурсов и сохраняется до начала следующего машинного цикла. Такая особенность интерфейса позволяет с помощью команды ХТНЛ БИС КР580ИК80А выполнять цикл чтение-модификация-запись, в течение которого системные ресурсы являются недоступными для ЦП микроЭВМ «Электроника 60». Возможность выполнения циклов имеет значение для организации обмена данными в многопроцессорных системах с использованием семифорных примитивов Р и V*.

Включение ОММПК в состав ресурсов ОС вычислительного комплекса (ВК) на базе МСВТ ряда «Электроника 60» может быть выполнено на одном из следующих трех уровней.

На уровне управления заданиями на ввод-вывод механизм межпроцессорного обмена данными можно использовать при соответствующей доработке управляющей системы серийной микроЭВМ. Синхронизация обмена сообщениями осуществляется с помощью прерываний от таймеров ОММПК и микроЭВМ соответственно. Таким образом реализуется единый, независимый от типа процессоров, механизм обмена данными в многопроцессорной системе, обеспечивающий повышение производительности за счет параллельного выполнения вычислений и процессов ввода-вывода при диспетчеризации заданий основным ЦП и одним или несколькими сопроцессорами ввода-вывода.

Более скромные результаты получаются при реализации взаимодействия ЦП и сопроцессора ввода-вывода на уровне программ драйверов конкретных устройств. Инициализация и завершение процессов ввода-вывода выполняются ЦП, а ОММПК обеспечивает непосредственное управление подключенными к нему УВВ. По окончании операции ввода-вывода сопроцессор прерывает работу ЦП и сообщает данные о текущем состоянии периферийного устройства. Производительность ВК повышается за счет параллельной работы ЦП и сопроцессора ввода-вывода. Доработке подлежат драйверы только тех устройств, управление которыми возлагается на ОММПК.

Обмен данными между ЦП и ОММПК может быть организован также с использованием почтовых ящиков специального вида — устройств, эмулирующих для ЦП микроЭВМ регистры стандартных контроллеров УВВ и обеспечивающих доступ ОММПК к потоку информации, которой ЦП обменивается с таким псевдоконт-

роллером. С помощью дополнительных аппаратных средств оказывается возможным использовать драйверы стандартных УВВ микроЭВМ «Электроника 60» для управления нестандартными внешними устройствами. Эмуляция контроллера стандартного УВВ микроЭВМ осуществляется средствами ПО ОММПК. Такое решение применено в выполняемой в настоящее время работе по созданию подключаемого к МПИ контроллера кассетного накопителя на магнитной ленте (НМЛ), эмулирующего контроллер устройства внешней памяти (УВП) на магнитной ленте типа СМ 5002.

Идея использования сопроцессоров типа ОММПК для решения задач управления в реальном масштабе времени может иметь дальнейшее развитие благодаря наличию собственных вычислительных ресурсов, в том числе и тех, которые подключаются к МПУ. Микроконтроллер обращается к МПИ только во время проверки состояния почтовых ящиков, чтения или записи данных в системном ОЗУ; подключение к МПИ нескольких сопроцессоров не приводит к заметным задержкам предоставления доступа к системной шине и снижению производительности ЦП. Перед разработчиками открывается возможность построения ВК, содержащих 16-разрядный ЦП (выполняющий трудоемкие вычислительные операции и управление заданиями) и ряд специализированных 8-разрядных сопроцессоров, решающих такие задачи, как управление УВП на НГМД и НМЛ, сбор и предварительная обработка аналоговой и цифровой информации, управление различными приборами, аппаратами и механизмами и т. п. В подобных иерархических системах для определенных классов задач может быть получено существенное сокращение времени реакции и увеличение производительности.

Помимо выполнения функций сопроцессора ввода-вывода ОММПК обладает всеми ресурсами, необходимыми для того, чтобы его можно было использовать в качестве встраиваемого контроллера устройств АВТ, ЦП недорогой 8-разрядной микроЭВМ, системы отладки и разработки ПО, элемента систем управления (СУ) технологическими процессами, транспортными средствами, робототехническими комплексами и т. п. В отличие от выпускаемых в настоящее время микроЭВМ и микроконтроллеров на БИС серии КР580 ОММПК позволяет при необходимости расширять вычислительные возможности построенной на его основе системы путем подключения к ней аппаратных ресурсов и ПО МСВТ «Электроника 60».

В настоящее время ОММПК реализован в виде макета, с помощью которого обрабатываются схемотехнические решения узлов сопряжения с УВВ и отлаживаются ПО. Накопленный опыт используется при разработке контроллера УВП на кассетном НМЛ.

Телефон для справок 419-09-94 (г. Киев).

Статья поступила 6 января 1986 г.

Из газеты «Вечерняя Москва», 10 сентября 1986 г.

«Большой зал Центрального лектория Всесоюзного общества «Знание» был переполнен. Тесно было на галерке, люди стояли в проходах, сидели на ступенях лестницы в амфитеатре. И все равно мест всем желающим не хватало».

Сюда на встречу с создателями новых отечественных компьютеров, которую организовала редакция журнала «Микропроцессорные средства и системы», пришли более тысячи ученых, инженеров, программистов, представителей промышленности...

* Для тех, кто пришел, но увы не смог попасть на встречу (а их оказалось около 500 человек), сообщаем, что семинар «Персональные компьютеры: серийные изделия отечественной промышленности» будет повторен 3 февраля 1987 г. в 12⁰⁰ (т. 923-00-19).

УДК 681.326

С. Б. Книгин

МНОГОГОЛОСЫЙ ЭЛЕКТРОМУЗЫКАЛЬНЫЙ СИНТЕЗАТОР

Цифровые электронные музыкальные синтезаторы (ЭМС) представляют собой микроЭВМ, периферийными устройствами которых являются клавиатура, пульт управления и программируемые узлы синтеза спектральных и временных характеристик звука.

Основная идея цифро-аналоговых электронных музыкальных синтезаторов эстрадного применения состоит

ЭМС, в которых каждой клавише соответствует свой канал синтеза, такие инструменты получили название многоголосых ЭМС. Число «голосов» в них обычно составляет 4...10. Каждый канал содержит чисто цифровые и аналоговые управляемые кодом узлы.

Предлагается описание цифро-аналогового многоголосого ЭМС на базе МП комплекта БИС КР580.

Характеристики ЭМС

| | |
|---|-----|
| Диапазон по клавиатуре, октав | 4 |
| Число каналов синтеза | 6 |
| Число тональных генераторов в канале | 2 |
| Погрешность установления частоты основных тонов не более, % от полутона | 2,9 |
| Время срабатывания клавиши не более, мс | 3,2 |
| Число оперативно выбираемых вариантов звучания | 48 |

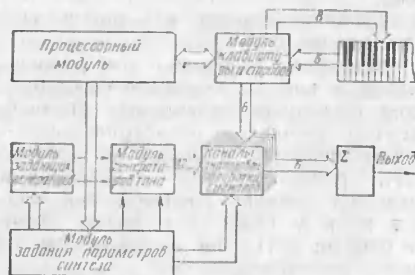


Рис. 1. Структурная схема ЭМС

в том, что с помощью специальных алгоритмов обработки сигналов от клавиатуры каждой нажатой клавише предоставляется во «временное пользование» канал синтеза звука, который в другой момент может предоставляться любой другой клавише. Поскольку реально число нажатых клавиш всегда меньше их общего числа, то, очевидно, это позволяет существенно уменьшить необходимое число каналов синтеза при незначительном сокращении функциональных возможностей инструмента. В отличие от полифонических

В структурной схеме ЭМС (рис. 1) процессорный модуль является ведущим узлом. Под его управлением работают модули клавиатуры и стробов, задания параметров синтеза и генераторов тона. Модуль клавиатуры связывает процессор с клавиатурой и формирует строб-сигналы для каждого из шести каналов. Модуль задания параметров синтеза вырабатывает вектор управляющих напряжений, определяющий значения программируемых параметров синтеза звука. Модуль генераторов тона из двух опорных частот вырабатывает шесть пар тональных сигналов.

Общие принципы синтеза звука и способы их реализации в данном ЭМС. Любому звуку соответствует неразрывная совокупность его амплитудной и спектральной характеристик, непрерывно изменяющихся во времени. Следовательно, задача состоит в получении комплекса признаков (являющихся функциями времени) — звуковысотного, спектрального и амплитудного. Соответствующий комплекс аппаратуры (управляемые генератор, фильтр, усилитель, а так-

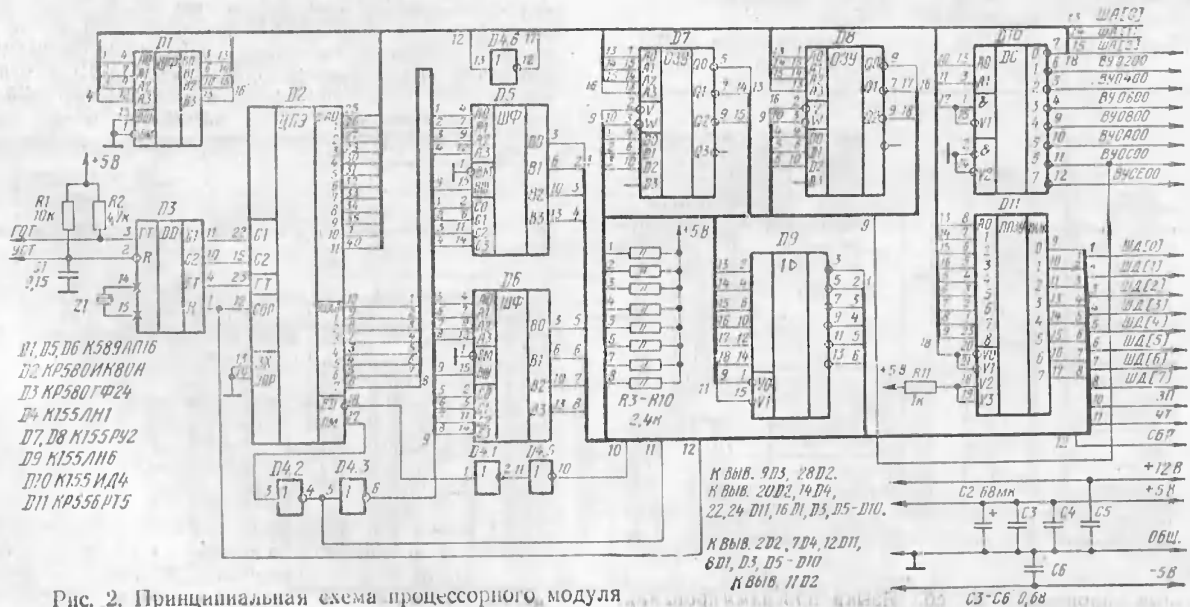


Рис. 2. Принципиальная схема процессорного модуля

же генераторы функций времени) входит в состав канала синтеза. Поскольку определяющим признаком музыкального звука является его высота (т. е. частота основного тона), управляемый генератор должен обладать достаточной точностью в широком диапазоне генерируемых частот. Этому требованию удовлетворяет управляемый кодом генератор на основе счетчика с переменным коэффициентом деления. Он легко реализуется на таймерах, входящих в МКК КР580. Таблица коэффициентов деления для всех 48 клавиш размещена в ПЗУ, где каждый коэффициент занимает два байта.

Полученный на выходе таймера меандр проходит дальнейшую обработку в канале синтеза для получения необходимых спектрально- и амплитудно-временных характеристик.

Генераторы функций времени запускаются строб-сигналом, который вырабатывается процессором для каждого канала на основании информации об изменении состояния клавиатуры.

Модуль процессора (рис. 2). В состав модуля кроме микропроцессора (D2), входят: генератор тактовых импульсов (D3), формирователи сигналов шины адресов, шины данных и шины управления, дешифратор адреса (D10), ОЗУ (D7, D8) и ПЗУ (D11). Генератор тактовых импульсов стабилизирован резонатором Z1 с частотой 13 000 кГц. Кроме того, элемент D3 формирует сигналы ГОТ и УСТ (начальная установка всех узлов схемы при включении питания). Внешние входы этих сигналов используются только при отладке схемы. Элементы D1 и D4.6 повышают нагрузочную способность тех линий шины адресов (ША), к которым требуется подключить более одного входа ТТЛ интегральных схем. Элементы D5 и D6 формируют сигналы двунаправленной шины данных (ШД). Направление передачи определяется сигналами

ПМ микропроцессора, который через элементы D4.2 и D4.3 поступает на входы ВШ (выбора шины) шинных формирователей (D5 и D6).

Оперативное запоминающее устройство емкостью 16 6-разрядных слов служит для хранения кодов клавиш, максимальное значение которых 2FH. Поэтому два старших разряда ШД в ОЗУ не используются.

Выходы ОЗУ подключаются к ШД через буферный элемент D9 с тремя состояниями выхода. Программируемое ПЗУ (D11) содержит программу ЭМС, таблицу частот тональных генераторов и таблицу оперативно выбираемых звучаний. Используемое адресное пространство микропроцессора разбито на восемь страниц по 512 байт. Такое разбиение осуществляется дешифратором (D10), входы которого подключены прямо к ША [9], ША [10] и через D4.6 к ША [11]. При этом обращение ко всем внешним регистрам происходит так же, как к ячейкам памяти. В большинстве страниц используется 1...6 адресов (табл. 1). Столь щедрое распределение адресного пространства имеет целью упростить программу и вполне допустимо во встраиваемых контроллерах.

Модуль клавиатуры и стробов (рис. 3) выполнен на основе ППА (D1). Все три канала ППА настроены для работы в режиме 0, причем каналы А и В на вывод, а канал С на ввод информации. Клавиатура представляет собой матрицу (8×6) нормально разомкнутых контактов, столбцы которой подключены к дешифратору D2, а строки ко входам мультиплексора D4. Диоды V1...V48 исключают взаимное влияние нажатых клавиш. Процессор последовательно загружает в порт А ППА (его адрес 0200H) коды клавиш от 00H до 2FH. Три младшие линии порта А преобразуются дешифратором D2 в позиционный код, а линии A3...A5 управляют работой мультиплексора D4. Если клавиша на пересечении выбранных столбца и строки нажата, то на выходе D4 появляется сигнал высокого уровня, который считывается процессором по линии С0. На основании этой информации может быть произведена установка в «1» одного из строб-сигналов СТР1...СТР6. Происходит это следующим образом.

По состоянию сигналов СТР1...СТР6 с помощью ППЗУ (D9) определяется номер свободного канала (канал считывается свободным, если соответствующий ему строб-сигнал имеет низкий уровень). Он считывается процессором с линий ШД0...ШД3 по сигналу ВУ0400H и выдается в порт В ППА, с линий которого поступает на дешифратор D5. Сигнал ЗП через интегрирующую цепь R4C1 поступает на вход разрешения дешифратора. В результате на соответствующем выходе D5 появляется отрицательный импульс, по ко-

Таблица 1

Распределение адресного пространства

| Поле адресов | Используемые адреса | Назначение |
|--------------|---------------------|---------------------------------------|
| 0000—01FF | 0000—01FF | ПЗУ программ |
| 0200—03FF | 0200—0203 | ППА модуля клавиатуры и стробов |
| 0400—05FF | 0400 | ПЗУ номера свободного канала |
| 0600—07FF | 0600—0603 | ППА модуля задания параметров синтеза |
| 0800—09FF | 0801—0806 | Регистры данных таймеров |
| 0A00—0BFF | 0A00 | РУС таймеров |
| 0C00—0DFF | 0C00—0C0F | ОЗУ |
| 0E00—0FFF | 0E00 | Сброс триггера «ПРОГРАММА» |

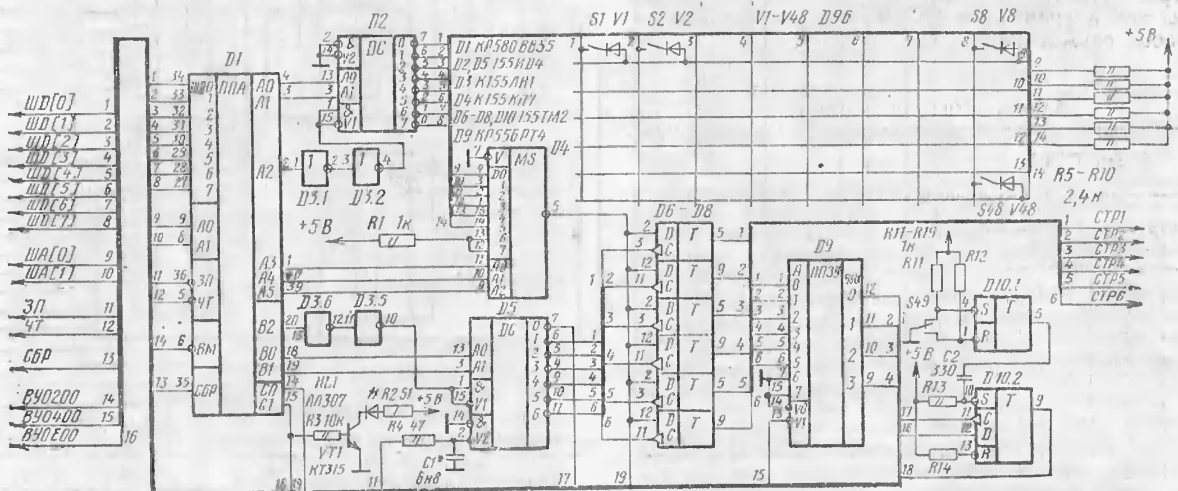


Рис. 3. Модуль клавиатуры и стробов

тому сигнал с выхода D4 переписывается в один из триггеров D6..D8. После этого в канал В заносится код 00H.

Сигнал Программа с кнопки S49 формируется триггером D10.1 и через дифференцирующую цепочку С2R13 поступает на вход установки триггера D10.2. Информация с его выхода считывается процессором по линии С1 ППА. Активный уровень на этой линии инициирует отработку подпрограммы выбора синтезируемого звучания, которая заключается в загрузке портов А, В и С ППА модуля выбора параметров синтеза данными из таблицы в соответствии с кодом нажатой клавиши.

После отработки подпрограммы происходит сброс триггера записью произвольного слова по адресу 0E00H. Светодиод HL1 служит для индикации режима Программа и расположен на передней панели ЭМС.

Возможный вариант прошивки ППЗУ (D9)

| | | | | | | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0000 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 4 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 5 |
| 0010 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 4 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 6 |
| 0020 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 4 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 5 |
| 0030 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 4 | 1 | 2 | 1 | 3 | 1 | 2 | 1 | 0 |

Появление на выходах D9 четырех нулей свидетельствует о том, что все каналы заняты. В этом случае процессор производит операции с нулевым каналом, который физически не существует, т. е. выполняет операции по обработке канальной информации вхолостую.

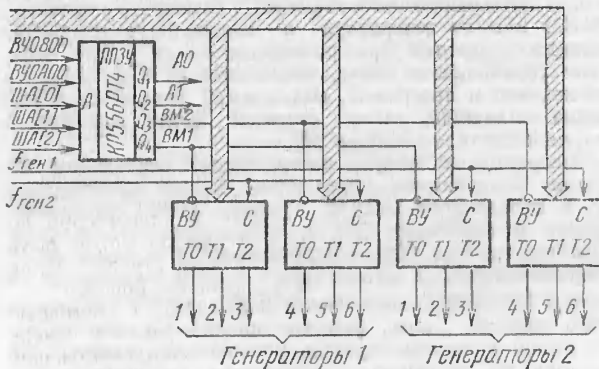


Рис. 4. Модуль генераторов тона

Модуль генераторов тона (рис. 4) выполнен на четырех таймерах КР580ВН53 [3], запрограммированных на работу в режиме 3 (генератор импульсов со скважностью 2). С помощью ППЗУ регистрам таймеров присваиваются адреса (табл. 2). Этот прием также, как и применение ППЗУ в модуле клавиатуры и стробов, служит для упрощения программы.

Таблица 2

Адреса регистров таймера

| Входы ППЗУ | | | | Выходы ППЗУ | | | | Регистр-приемник | |
|------------|--------|----|----|-------------|-----|-----|----|------------------|-----|
| ВУ0800 | ВУ0A00 | A2 | A1 | A0 | ВМ2 | ВМ1 | A1 | | A0 |
| 1 | 0 | X | X | X | 0 | 0 | 1 | 1 | РУС |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | PD0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | PD1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | PD2 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | PD0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | PD1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | PD2 |
| 1 | 1 | X | X | X | 1 | 1 | 1 | 1 | — |

Следует отметить, что решение части задач аппаратными средствами необходимо не только для уменьшения объема ППЗУ, но и для уменьшения времени «срабатывания» клавиш, которое зависит от продолжительности одного полного прохода цикла программы и совместно с механическими характеристиками клавиатуры определяет ее качество. Это время не должно быть более 6 мс [1].

Модуль задания параметров на основе ППА КР580ВВ55. Все порты модуля настроены на вывод в режиме 0. Совокупность линий портов А, В и С — это один 24-разрядный выходной регистр, каждый разряд которого можно использовать для управления параметром, имеющим два значения (например, скважность тонального сигнала 2 или 4), а группа разрядов — для выбора значений управляющего напряжения (посредством ЦАП или аналогового коммутатора). Конкретное назначение каждого разряда определяется при выборе схемы каналов синтеза и набора управляемых параметров.

Все модули выполнены на двухсторонних печатных платах, соединенных между собой жгутом. В настоящее время разрабатывается ЭМС с одноплатным контактным роликом.

Программное обеспечение ЭМС.

При включении питания ЭМС происходит сброс процессора и ППА. С адреса 0000H начинается отработка программы: настройка портов и таймеров для работы в соответствующих режимах и заполнение ячеек ОЗУ последовательностью кодов от 31H до 36H. Это необходимо для исключения «молчащих» клавиш, которые могут появиться при случайной установке кодов в ОЗУ после подачи питания (рис. 5).

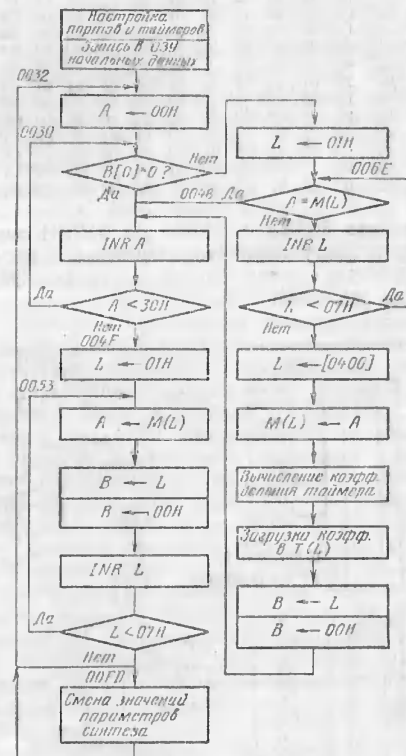


Рис. 5. Алгоритм работы ЭМС: А — код клавиши (состояние порта А ППА в модуле клавиатуры и стробов); В — код строб-сигнала (состояние порта В ППА); L — номер канала синтеза; M(L) — ячейка ОЗУ, соответствующая каналу с номером L; T(L) — регистр данных таймера, соответствующий каналу с номером L.

Основная часть программы начинается с опроса нулевой клавиши (на клавиатуре она первая справа). Если она не нажата, то опрашивается следующая клавиша и т. д. Если очередная клавиша нажата, то выполняется часть программы, обрабатывающая коды нажатых клавиш. Она определяет, содержится ли код данной клавиши в памяти одного из каналов, если нет, ставит в соответствие ей свободный канал синтеза. Это достигается вводом номера свободного канала из ПЗУ модуля клавиатуры и стробов, загрузкой кода клавиши в ячейку ОЗУ, соответствующую этому каналу; выбором из таблицы и загрузкой в таймер двухбайтного кода, определяющего частоту тонального сигнала и установкой активного уровня канального строб-сигнала. Далее осуществляется переход к опросу следующей клавиши. После опроса всех 48 клавиш устанавливаются строб-сигналы всех каналов в соответствии с состоянием клавиш. После этого алгоритм повторяется с опроса нулевой клавиши. Если сигнал Программа принимает активный уровень, то после очередного прохода выполняется программа смены звучания, состоящая в поиске нажатой клавиши и загрузке ППА модуля задания параметров синтеза соответствующими ей кодами из таблицы.

Содержимое ПЗУ программ

| ADR. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0000 | 3E | 89 | 32 | 03 | 02 | 3E | 80 | 32 | 03 | 06 | 3E | 36 | 32 | 00 | 0A | 3E |
| 0010 | 76 | 32 | 00 | 0A | 3E | B6 | 32 | 00 | 0A | 3E | 31 | 32 | 06 | 0C | 3C | 32 |
| 0020 | 01 | 0C | 3C | 32 | 02 | 0C | 3C | 32 | 03 | 0C | 3C | 32 | 04 | 0C | 3C | 32 |
| 0030 | 05 | 0C | 3A | 02 | 02 | E6 | 02 | C2 | FD | 00 | 0E | 00 | 79 | 32 | 00 | 02 |
| 0040 | 3A | 02 | 02 | E6 | 01 | C2 | 6A | 00 | 0C | 79 | FE | 30 | DA | 3C | 00 | 26 |
| 0050 | 0C | 2E | 01 | 7E | 32 | 00 | 02 | 7D | 32 | 01 | 02 | 3E | 00 | 32 | 01 | 02 |
| 0060 | 2C | 7D | FE | 07 | DA | 53 | 00 | C3 | 32 | 0A | 26 | 2E | 2E | 01 | 7E | F6 |
| 0070 | 3F | B9 | CA | 48 | 00 | 2C | 7D | FE | 07 | DA | 6E | 00 | 3A | 00 | 04 | 6F |
| 0080 | 26 | 0C | 71 | 26 | 08 | 16 | 09 | 79 | 07 | C6 | 9D | 5F | 1A | 77 | 1C | 7A |
| 0090 | 77 | 7D | 32 | 01 | 02 | 3E | 00 | 32 | 01 | 02 | C3 | 48 | 00 | D8 | 06 | 75 |
| 00A0 | 06 | 19 | 06 | C1 | 05 | 6E | 05 | 20 | 05 | D7 | 04 | 91 | 04 | 50 | 04 | 12 |
| 00B0 | 04 | D7 | 03 | A0 | 03 | 6C | 03 | 3B | 03 | 0C | 03 | E1 | 02 | B7 | 02 | 90 |
| 00C0 | 02 | 6B | 62 | 49 | 02 | 28 | 02 | 09 | 02 | FC | 01 | D9 | 01 | B6 | 01 | 8D |
| 00D0 | 01 | 86 | 01 | 70 | 01 | 51 | 00 | 4F | 01 | 36 | 01 | 24 | 01 | 14 | 01 | 04 |
| 00E0 | 01 | F6 | 00 | E8 | 00 | DB | 00 | CF | 00 | C3 | 00 | B8 | 00 | AE | 00 | A4 |
| 00F0 | 00 | 9B | 00 | 92 | 00 | 8A | 00 | 82 | 00 | 7B | 00 | 74 | 00 | 0E | 00 | 79 |
| 0100 | 32 | 00 | 02 | 3A | 02 | 02 | E6 | 01 | C2 | 0F | 01 | 0C | C3 | FF | 00 | 79 |
| 0110 | E6 | 3F | 51 | C6 | 40 | 4F | 06 | 01 | 6A | 32 | 00 | 06 | 79 | C6 | 40 | 4F |
| 0120 | 0A | 32 | 01 | 06 | 79 | C6 | 04 | 0A | 32 | 02 | 06 | 06 | 00 | 7A | 32 | 00 |
| 0130 | 02 | 3A | 02 | 02 | E6 | 01 | C2 | 31 | 01 | 32 | 00 | 0E | C3 | 3A | 00 | FF |

Часть адресов ПЗУ с 0140H по 01FFH зарезервирована под таблицу вариантов звучания и программируется в процессе окончательной настройки ЭМС после отработки звучаний.

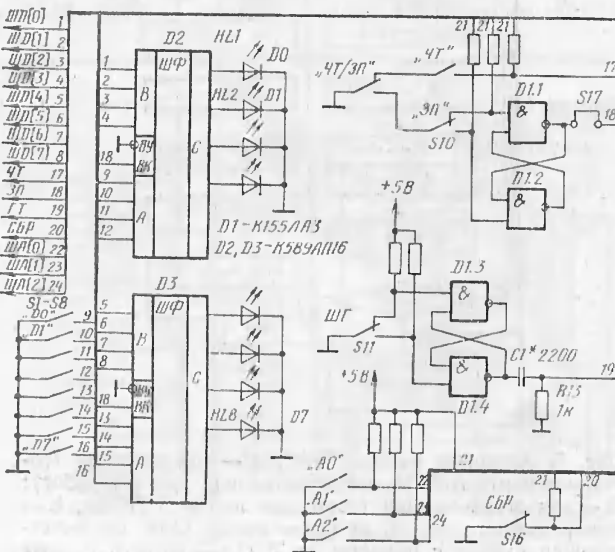


Рис. 6. Принципиальная схема отладочного модуля

Настройка цифровой части ЭМС производится с помощью простейшего отладочного модуля (рис. 6), состоящего из тумблерного регистра данных S1...S8, регистра индикации на светодиодах HL1...HL8, шинных формирователей D2 и D3, триггеров на элементах D1.1...D1.4 и ряда коммутационных элементов. Входы В ШФ (D2, D3) образуют двунаправленную шину данных, на которую можно вывести информацию с тумблерного регистра S1...S8 и прочитать ее на светодиодах HL1...HL8. Триггер на элементах D1.1 и D1.2 устраняет дребезг контактов кнопки S10 ЗП. Триггер на элементах D1.3 и D1.4 совместно с кнопкой S11 ШГ и дифференцирующей печкой C1R13 организует пошаговый режим работы процессора.

Отладочный модуль позволяет настраивать отдельные модули и контроллер в целом.

Проверяемый периферийный модуль присоединяется к отладочному с помощью короткого шлейфа. Затем, руководствуясь описаниями работы модуля и БИС [2, 3], средствами отладочного модуля, записывается и читается необходимая информация на шине данных. Сигналы внутри модуля контролируются логическим пробником. Таким образом, можно устранить ошибки в монтаже и убедиться в правильной работе модуля. Для модулей, содержащих ППА, необходимо до включения питания разомкнуть тумблер S16 СБР и замкнуть его через 1—2 с после включения питания.

Модуль генераторов тона проверяют, подавая на входы синхронизации таймеров сигнал с частотой 30...50 кГц от генератора и контролируя изменение выходной частоты при различных кодах. При этом, если управляющие слова, записанные в РУС таймеров те же, что в программе, код должен заноситься в два стаба (младший, затем старший байт) или просто двумя нажатиями кнопки ЗП.

Процессорный модуль настраивается следующим образом. Соединяются линии шины данных процессорного и отладочного модуля. Тумблер ЗП/ЧТ устанавливается в положение ЗП, а тумблерами S1...S8 устанавливается код 00H, соответствующий команде NOP. Переключатель S17 в отладочном модуле удаляется, а входы ВК ШФ подключаются к выходу ЧТ процессорного модуля. Далее, включив питание, можно осциллографом проконтролировать правильность работы процессора по описанию и временным диаграммам [4]. Затем устанавливается на плату ПЗУ КР556РТ5 с программой (желательно установить его на панельке), в процессорном модуле ставится резистор R2 и к входу ГОТ подключается выход ГТ отладочного модуля. В отладочном модуле восстанавливается переключатель S17, а тумблер ЧТ/ЗП ставится в положение ЧТ. Контролируя состояние шины данных по светодиодам HL1...HL8, подбирают емкость C1 в отладочном модуле так, чтобы при каждом нажатии кнопки ШАГ выполнялась только одна команда.

Теперь можно выполнить программу в пошаговом режиме и проверить правильность работы всего контроллера ЭМС в целом. Если ошибок в работе не обнаружено, то отладочный модуль отсоединяется, а резистор R2 устанавливается на свое место. Подключив клавиатуру и нажимая на ней одну две и так далее клавиши одновременно, контролируют появление строб-сигналов и изменение выходных частот таймеров. Если при какой-либо комбинации клавиш на одном или нескольких выходах строб-сигналов появляются импульсы, нужно подобрать емкость C1 в модуле клавиатуры и стробов. На этом отладку цифровой части ЭМС можно считать законченной.

К недостаткам ЭМС следует отнести небольшой для многоголосого ЭМС диапазон по клавиатуре и невозможность перестройки программ синтеза звука исполнителем.

Первый недостаток носит чисто конструктивный характер. Как схема, так и быстроедействие программы позволяют увеличить количество клавиш до 60. Для

этого достаточно задействовать входы 12 и 13 элемента D4 в модуле клавиатуры и стробов. Количество каналов также можно увеличить, но для этого нужно заменить дешифратор D5 на четырехходовой, увеличить количество триггеров, объем ПЗУ D9 и внести изменения в прошивку ПЗУ программ: число клавиш занести в ячейку с адресом 004В; число каналов плюс 1 — в ячейки с адресами 0063 и 0078.

Второй недостаток также можно устранить, но это приведет к существенному усложнению модуля задания параметров синтеза. Как показывает практика, даже имея ЭМС с неограниченными возможностями синтеза звука, музыкант использует не более 10..15 особо понравившихся звучаний (особенно популярны «аккордеон», «глубная гармоника», «клавесин», различные имитации струнных и духовых инструментов). Поэтому при удачном подборе имеющихся 48 звучаний, учитывая возможность их оперативной смены (что является несомненным преимуществом описанного ЭМС перед аналоговыми), инструмент придется по душе как любителям, так и профессиональным исполнителям.

Телефон для справок: 77 85 43 (г. Киров).

ЛИТЕРАТУРА

1. Пат. 3 986 423 (США).
2. Зеленко Г., Панов В., Попов С. Радиолобителю о микропроцессорах и микроЭВМ. Модуль программатора ППЗУ // Радио.—1983.— № 6.
3. Торгов Ю. И. Программируемый таймер КР580ВН53 и его применение // Микропроцессорные средства и системы.—1984.— № 1.— С. 77—84.
4. Зеленко Г., Панов В., Попов С. Радиолобителю о микропроцессорах и микроЭВМ. Процессорный модуль // Радио.—1983.— № 2.— С. 41—43.
5. Коффри Дж. Технические средства микропроцессорных систем. Практический курс: Пер. с англ.— М.: Мир, 1983.
6. Григорян В., Печатнов Б., Сабуров С., Сорокин С. Узлы ЭМС // Радио.—1981.— № 4.
7. Маргулис А., Парыгин Ю. Делители частоты многоголосого ЭМИ // В помощь радиолюбителю.— Вып. 79.— С. 10.

Статья поступила 15 марта 1986 г.

УДК 681.335.2

И. И. Дубровский, Л. Н. Финякин, В. В. Кафаров УСТРОЙСТВО ПАРАЛЛЕЛЬНОГО ВВОДА АНАЛОГОВОЙ ИНФОРМАЦИИ В МИКРОЭВМ

Микропроцессорные автоматические системы контроля в составе автоматизированных систем научных исследований обеспечивают обработку, промежуточное хранение информации и выдачу ее на внешние носители (магнитные диски, магнитные ленты, перфоленты или на печать). Примером связи аналоговых источников информации с микроЭВМ может быть модуль аналогового ввода в микроЭВМ [1], состоящий из аналого-цифрового преобразователя (АЦП), схемы сопряжения АЦП с общей магистралью микроЭВМ, называемой интерфейсом с тремя адресуемыми регистрами: регистром номера канала, буферным регистром данных и регистром состояния.

Сбор и обработка информации значительно упрощаются при параллельном вводе ее в микроЭВМ «Электроника ДЗ-28» (рис. 1). В состав предложенной схемы вошли четыре корпуса (DD1.1...DD4.2) двоянных одновибраторов K155AГ3, которые позволили за счет включения в их внешние времязадающие цепи резистивных и емкостных датчиков организовать параллельный ввод информации в микроЭВМ по шине Ввод.

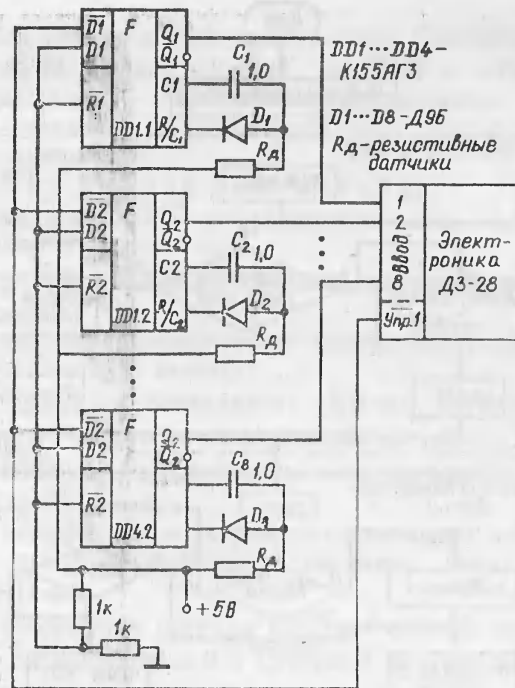


Рис. 1. Принципиальная схема устройства параллельного ввода аналоговых сигналов в микроЭВМ

Работа каждого из восьми одновибраторов состоит в том, что они преобразуют аналоговую информацию в эквивалентный ей широко-импульсный сигнал (ШИС) в некоторые дискретные моменты времени [2]. Наличие в «Электронике ДЗ-28» 8-разрядного регистра Ввод и возможности программным путем контролировать переход каждого разряда из исходного состояния, которое может быть либо «Лог. 1», либо «Лог. 0», в противоположное послужили основой для разработки схемы параллельного ввода аналоговой информации в микроЭВМ.

В зависимости от значений мгновенных величин внешней RC-цепочки каждый из одновибраторов формирует измерительный ШИС большей или меньшей длительности; момент начала формирования задается либо как «прерывание» от внешнего таймера (на рис. 1 не показан), либо командой микроЭВМ, выполняющей циклы заданной длительности. Началом измерительного цикла может быть некоторая адресная комбинация, передаваемая по шине Управление. Этой командой запускаются одновременно восемь одновибраторов, после чего микроЭВМ приступает к циклическому анализу состояния шины Ввод. Прохождение каждого цикла ввода байта с шины Ввод и анализ каждого разряда введенного байта сопровождается накоплением суммы в счетчике, возрастающей с постоянной скоростью. Переход некоторого разряда из исходного в противоположное состояние служит признаком завершения ШИС и вызывает пересылку текущего значения суммы из рабочего регистра в регистр хранения.

Главной отличительной особенностью программно-технических средств является то, что преобразование осуществляется сразу над группой данных, получаемых от первичных преобразователей в форме ШИС, по программе, приведенной на рис. 2. Программа состоит из четырех модулей.

Модуль 1 (Подготовка) заносит необходимые числовые и адресные константы в соответствующие регистры.

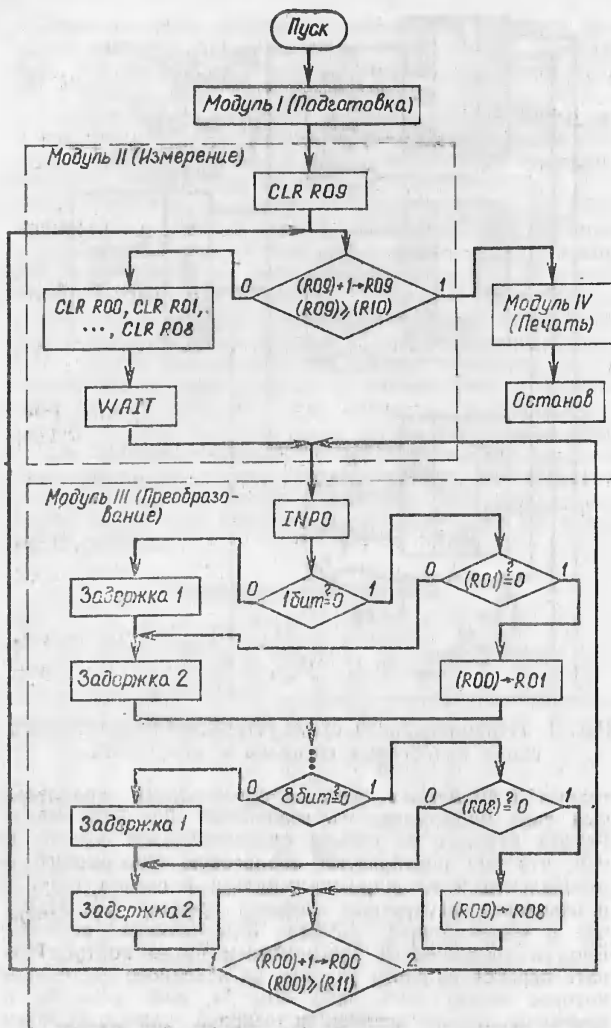


Рис. 2. Блок-схема программы обработки аналоговых сигналов от группы первичных преобразователей-датчиков:

R00, R01, ..., R01 — рабочие регистры ЭВМ; (R00), (R01), ..., (R01) — содержимое регистров R00, R01, ..., R01; условия: «1» — ДА, «0» — НЕТ

ры микроЭВМ, среди них: N_{max} — максимальное количество циклов измерения (число замеров в коде эксперимента) и M_{max} — максимальное количество счетных программных циклов на протяжении одного измерения.

Модуль II (Измерение) осуществляет привязку начала измерений к моменту времени $t_n=0$ и подготавливает регистр счета измерительных циклов, следующие в дискретные моменты времени с тем или иным шагом дискретизации (периодичностью) Δt в зависимости от динамики исследуемого процесса. После проверки выполнения условия $N_{текущ} \leq N_{max}$, т. е. $(R09) \geq (R10)$, микроЭВМ переходит в режим ожидания внешнего прерывания, при поступлении которого управление передается программному модулю III.

Модуль III (Преобразование) преобразует аналоговые величины, закодированные в ШИС, в числовую форму для последующей передачи этих значений в рабочую область ОЗУ.

Модуль IV выдает цифровую информацию на внешний носитель, например на печать.

В основу третьего модуля положен способ программного преобразования аналоговых величин в их цифровые эквиваленты при представлении контролируемых физических величин в виде ШИС, где информация заложена в его длительности. Это преобразование в цифровой эквивалент осуществляется путем заполнения ШИС счетными импульсами, в качестве которых используются программные циклы с постоянной длительностью: сложение содержимого одного из рабочих регистров с константой (единицей) непосредственно в ЭВМ. В данном случае отпадает необходимость в сложном и дорогостоящем оборудовании: аналого-цифровом преобразователе и устройстве сопряжения этого АЦП с микроЭВМ. Чтобы обеспечить высокую метрологическую точность преобразования аналоговой величины в цифровую, используются две программные задержки: Задержка 1, эквивалентная по длительности логической операции проверки на нуль содержимого регистра R_i, и Задержка 2, эквивалентная пересылке числа из регистра в регистр.

При переходе к модулю III управление передается команде INPO, по которой информация от датчиков поступает через шину Ввод в рабочий регистр. Следующей командой осуществляется анализ состояния первого разряда этого регистра. Предположим, что в первых циклах работы этой подпрограммы изменения в исходном состоянии регистра Ввод не происходят. Поэтому программа выходит на обработку программных задержек (Задержка 1 и Задержка 2); затем по аналогичным алгоритмам анализируются остальные биты. После проверки всех разрядов содержимое регистра $(R00)+1 \rightarrow R00$ увеличивается, и оно сравнивается со значением M_{max} : когда содержимое регистра R00 меньше M_{max} , выполняется переход опять на команду INPO.

В случае, если при выполнении i-го цикла ввода обнаружено изменение исходного состояния первого разряда регистра Ввод, то произойдет переход не к обработке задержек, а к анализу состояния рабочего регистра, закрепленного за данным разрядом — R01. Если до этого информация в него не заносилась, т. е. $(R01)=0$, то содержимое из регистра R00 пересылается в регистр R01, т. е. фиксируется количество циклов, выполненных к этому моменту. Затем анализируется состояние второго разряда и т. д. В следующем (i+1)-м цикле ввода состояние первого разряда будет подтверждено. Поэтому при его анализе, но с содержимым регистра R01, отличным от нуля, пересылка числового значения из R00 в R01 будет запрещена. Чтобы обеспечить постоянство длительности программных циклов, необходимо отработать Задержку 2 с последующим выходом на анализ второго разряда и т. д. По завершении анализа восьмого разряда к содержимому R00 прибавляется очередная единица и производится сравнение значения $(R00)$ с M_{max} , т. е. $(R00)$ и $(R11)$. Измерительные циклы продолжают до тех пор, пока R00 не станет равным M_{max} .

В результате в каждом из регистров R01, R02, ..., R08 будет записано число, пропорциональное длительности соответствующего ШИС, сформированного тем или иным одновибратором. По завершении измерительного цикла информация в числовой форме передается из регистров R01, R02, ..., R08 в рабочую область ОЗУ.

После перезаписи информации в стек ОЗУ осуществляется возврат в модуль II (прибавляется очередная единица, очищаются рабочие регистры и т. д.).

В предложенном устройстве параллельного ввода аналоговых сигналов в микроЭВМ во внешних времязадающих цепях установлены терморезисторы. Так как одновибраторы К155АГ3 — стробируемые, число групп датчиков (по 8 штук в каждой) можно легко расширить схемно-программным путем.

При увеличении числа точек измерения следует учитывать, что быстродействие системы невелико — обус-

жвание группы из восьми одновибраторов при применении ЭВМ типа «Электроника ДЗ-28» требует приблизительно 1,5 с (машинный такт 2 мкс), если $M_{max} = 500$ (что обеспечивает точность преобразования 0,2 %).

Разработанная система предназначена для изучения процесса смешивания жидкостей в лабораторном аппарате путем регистрации теплового поля в объеме. Терморезисторы помещены во внутреннюю полость аппарата для регистрации температурных изменений в различных его точках при нанесении внешнего возмущения в виде теплового импульсного (или ступенчатого) воздействия. По полученным числовым значениям производилась оценка структуры потоков, определялись размеры зон для декомпозиционного математического описания гидродинамики аппарата и последующего синтеза полной его модели согласно основным принципам системного анализа [3].

ЛИТЕРАТУРА

1. Фролов Г. И., Гембицкий Р. А. Автоматизированные системы контроля объектов. — М.: Высшая школа, 1984. — 69 с.
2. Дубровский И. И., Финякин Л. Н. Использование программных средств при сопряжении микроЭВМ с химико-технологическим объектом // В кн.: Тез. докл. Всесоюз. науч. конф. «КХТП-1». — М.: МХТИ, 1984. — 216 с.
3. Кафаров В. В. Методы кибернетики в химии и химической технологии. — М.: Химия, 1985. — 448 с.

Статья поступила 16 июля 1985 г.

УДК 681.326

С. И. Васильев, В. Б. Леонов

СОПРЯЖЕНИЕ МИКРОЭВМ «ЭЛЕКТРОНИКА ДЗ-28» С МПК БИС СЕРИИ КР580

Известно, что на основе микропроцессорного комплекта (МПК) серии КР580 можно создавать необходимые периферийные блоки [1, 2]. Разработано устройство преобразования (УП), позволяющее использовать микросхемы серии КР580 совместно с микроЭВМ «Электроника ДЗ-28».

При разработке УП требовалось обеспечить функционирование периферийных устройств, уже имеющих у «ДЗ-28». Кроме того, желательно, чтобы в те моменты времени, когда «ДЗ-28» не выполняет операций ввода-вывода, магистраль МПК КР580 оставалась функционально отключенной от ЭВМ. УП можно использовать для отладки микропроцессорных систем, построенных на основе МПК КР580. УП не содержит узлов для поддержки работы в режиме прерываний, так как у «ДЗ-28» имеется четыре входа запроса прерываний, на которые могут быть поданы сигналы запроса от устройств, выполненных на основе микросхем серии КР580.

Магистраль ввода-вывода «Электроника ДЗ-28» состоит из трех 8-разрядных шин (ВВОД, ВВЫВ, УПР — адресная магистраль), линий сигналов синхронизации (ВВ, СИМ, СИП) и четырех линий запроса прерываний (ПР1, ПР4, ПР2, ПР8). Все сигналы переда-

ются низким логическим уровнем. Сигналы ВВ и СИМ выдает «ДЗ-28», а СИП и сигналы прерываний — периферийное устройство.

Порядок функционирования магистрали следующий.

При выводе информации:

на шине УПР выставляется адрес периферийного устройства;

на шине ВВВ устанавливается байт информации;

«ДЗ-28» устанавливает сигнал СИМ=0, что соответствует выводу;

«ДЗ-28» устанавливает ВВ=0, сигнализируя о готовности информации на шинах УПР и ВВВ, и переходит в режим ожидания сигнала СИП=0;

периферийное устройство принимает информацию и сигнализирует об этом, выставляя СИП=0;

по приходу сигнала СИП=0 «ДЗ-28» снимает сигналы ВВ=0 и СИМ=0, в ответ на это устройство должно снять сигнал СИП=0; на этом цикл вывода заканчивается.

При вводе информации:

на шине УПР выставляется адрес периферийного устройства;

сигнал СИМ=1, что соответствует вводу;

«ДЗ-28» устанавливает ВВ=1, сигнализируя о готовности принять информацию и готовности информации на шине УПР, и переходит в режим ожидания сигнала СИП=0;

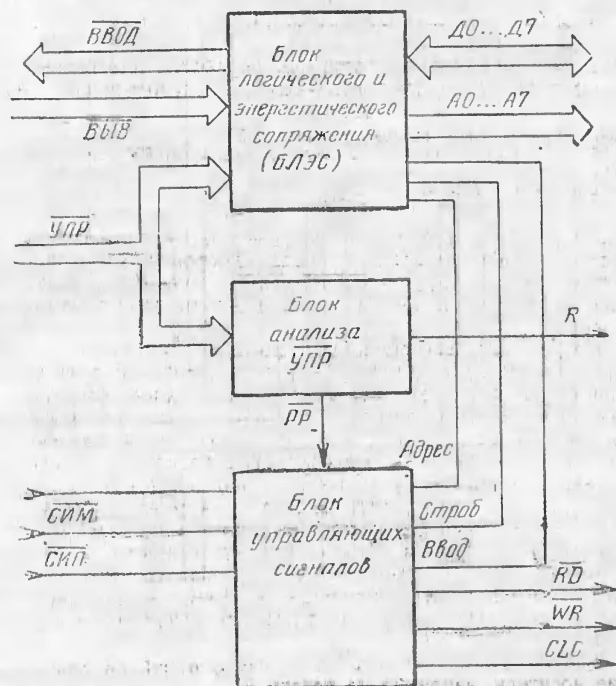


Рис. 1. Структурная схема УП

периферийное устройство выставляет информацию на шине ВВОД и сигнализирует об этом, выставляя СИП=0;

по приходу сигнала СИП=0 «ДЗ-28» принимает информацию с шины ВВОД и после этого выставляет сигнал ВВ=1, в ответ на это периферийное устройство должно снять сигнал СИП=0; на этом цикл ввода заканчивается.

Устройство преобразования (рис. 1) согласует вышеописанный протокол обмена с протоколом МПК КР580.

Блок управляющих сигналов (рис. 2, а) получает от «ДЗ-28» сигналы ВВ и СИМ, затем вырабатывает сигнал СИП для «ДЗ-28» и сигнал записи WR и чтения RD для магистрали МПК КР580, а также сигналы Адр, Строб и Ввод, управляющие работой БЛЭС.

БЛЭС (рис. 2, б) преобразует сигналы двух однонаправленных шин «ДЗ-28» ВВОД и ВЫВ в сигналы двунаправленной шины данных D0...D7, а сигналы шины УПР в сигналы шины адреса A0...A7. Блок также осуществляет временные преобразования, необходимые для согласования асинхронной магистрали «ДЗ-28» и синхронной магистрали МПК КР580.

Блок анализа УПР (рис. 2, в) при состояниях старших четырех разрядов магистрали УПР, не равных 0Н, 3Н, ЕН, FN, вырабатывает сигнал разрешения работы (PP), который

поступает на блок управляющих сигналов. Наличие стандартных устройств ввода-вывода вызывает необходимость запретить работу УП при УПР=Е0Н...FNH. При старших четырех разрядах УПР=3Н вырабатывается сигнал сброса (R) для магистрали МПК КР580.

Если сигнал PP=0, то с приходом от «ДЗ-28» сигнала ВВ на счетчик D6 перестает поступать сигнал сброса, а на выходе D4.1 при условии СИМ=1 (ЭВМ производит ввод информации) формируется сигнал Ввод. Импульсы с выхода генератора D2.5, D2.6, D3.1, D10 поступают на вход микросхемы D4.2 и на магистраль в качестве тактовых импульсов CLC. На другой вход D4.2 в исходном состоянии поступает единичный уровень. По мере поступления тактовых импульсов счетчик D6 последовательно проходит через состояния 0...7. При этом появляются нулевые уровни на соответствующих выходах дешифратора D7.

В состояниях 2—6 счетчика на выходе D8 появляется единичный уровень, который через инвертор поступает на БЛЭС, разрешая выдачу адреса A0...A7 и обмен информацией по

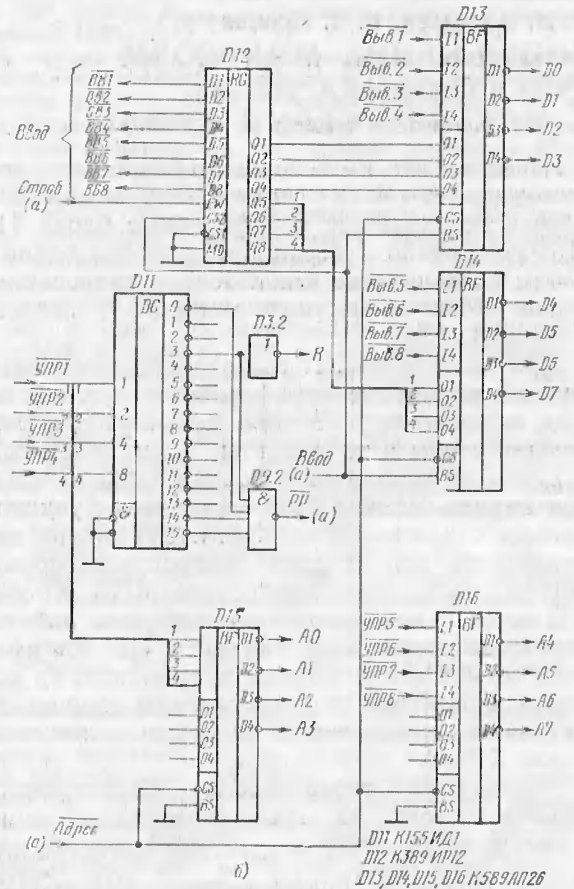
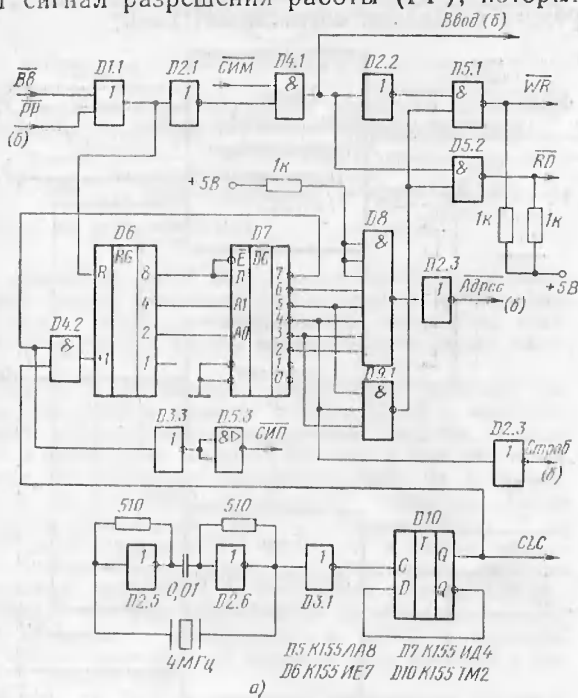


Рис. 2. Принципиальная схема блока управляющих сигналов (а), БЛЭС и блока анализа УПР (б)

шине данных D0...D7. Когда счетчик находится в состояниях 3...5, на выходе D9.1 формируется единичный уровень, поступающий на входы D5.2, D5.1. При этом разрешается выдача сигналов \overline{RD} , \overline{WR} .

В состоянии 4 счетчика формируется сигнал Строб, по которому в БЛЭС производится запись информации в буферный регистр D12. Следует отметить, что в те моменты времени, когда «ДЗ-28» не производит операций ввода-вывода, шина ВВОД используется для передачи служебной информации, поэтому в качестве буферного регистра используется схема с тристабильным выходом.

В состоянии 7 счетчика D6 на выход элемента D4.2 поступает нулевой уровень. Этим запрещается поступление тактовых импульсов на вход счетчика. На выходе D5.3 в состоянии 7 формируется сигнал СИП. «ДЗ-28», получив сигнал СИП, снимает сигнал ВВ. При этом на вход начальной установки D6 поступает

единичный уровень и счетчик устанавливается в нулевое состояние. На этом цикл обмена заканчивается.

Блок анализа УПР выполнен на дешифраторе D11 и элементах D8.2, D9.2. Сигнал с выхода D3.2 поступает на магистраль МПК КР580 в качестве сигнала сброса R.

Описанное устройство применялось для сопряжения с устройствами управления высокочастотными ключевыми схемами, графопостроителем Н306 (через интерфейсную плату от ЭВМ «Искра 226») и модулем связи с линией «токовая петля 20 мА» и «стык С2».

Телефон для справок: 139-20-94 (г. Москва).

ЛИТЕРАТУРА

1. Хоуп Г. Проектирование цифровых вычислительных устройств на интегральных схемах.— М.: Мир, 1984.— 400 с.
2. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.— 270 с.

Статья поступила 28 октября 1985 г.

УДК 681.325.54

Д. И. Панфилов, С. Г. Шаронин, С. Е. Яковлев

ОРГАНИЗАЦИЯ ОБМЕНА ИНФОРМАЦИЕЙ С МИКРОЭВМ В ПАРАЛЛЕЛЬНОМ КОДЕ

Описание устройства

Устройство для исследования организации обмена информацией с микроЭВМ в параллельном коде собрано на печатной плате размером 240×120 мм и имеет выведенные на внешний разъем 16-разрядную магистраль адресов (МА), 8-разрядную магистраль данных (МД) и 5-разряд-

ную магистраль управления (МУ) (рис. 1). В состав устройства входят:

- БИС программируемого параллельного интерфейса (ППИ) КР580ВВ55 [1];
- дешифратор адреса (ДША);
- схема индикации управляющего слова (СИУС), позволяющая отображать на светодиодах содержимое

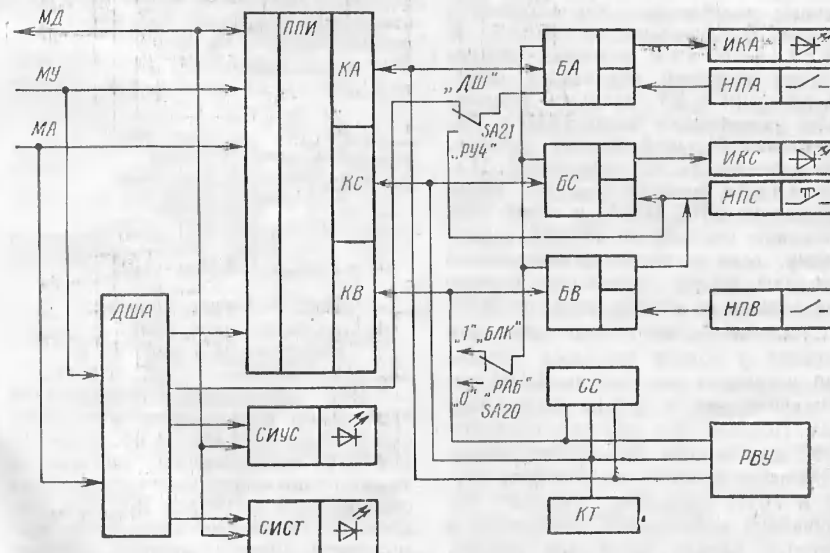


Рис. 1. Структурная схема параллельного интерфейса

внутреннего регистра управления ППИ во всех режимах его работы; схема индикации состояния внутренних триггеров разрешения прерываний (СИСТ);

буферы каналов А, В и С (БА, БВ и БС), позволяющие подключать наборные поля, схемы индикации каналов и различные ВУ;

схемы индикации каналов А и С (ИКА, ИКС);

наборные поля каналов А, В и С (НПА, НПВ, НПС) для имитации управляющих сигналов БИС КР580ВВ55 и установки входной информации;

схема формирования управляющих сигналов для сопряжения двух микроЭВМ с помощью данного устройства (СС);

разъемы для подключения ВУ (РВУ);

переключатели SA20 и SA21, выполняющие следующие функции: SA20 — подключение к БИС ППИ наборных полей и индикаторов каналов А, В, С через буферы БА, БВ, БС (положение «РАБ»), блокировка буферов при подключении к БИС ВУ (положение «БЛК»); SA21 — управление записью данных из канала А БИС ППИ в регистр ИКА по сигналу с дешифратора (положение «ДШ») или по сигналу с наборного поля канала С (положение «РУЧ»);

контрольные точки (КТ), на которые выведены все разряды канала С. Они могут быть использованы для подключения осциллографа при исследовании временных диаграмм работы БИС ППИ.

Принципиальная электрическая схема устройства сопряжения приведена на рис. 2 (без схемы сопряжения двух микроЭВМ).

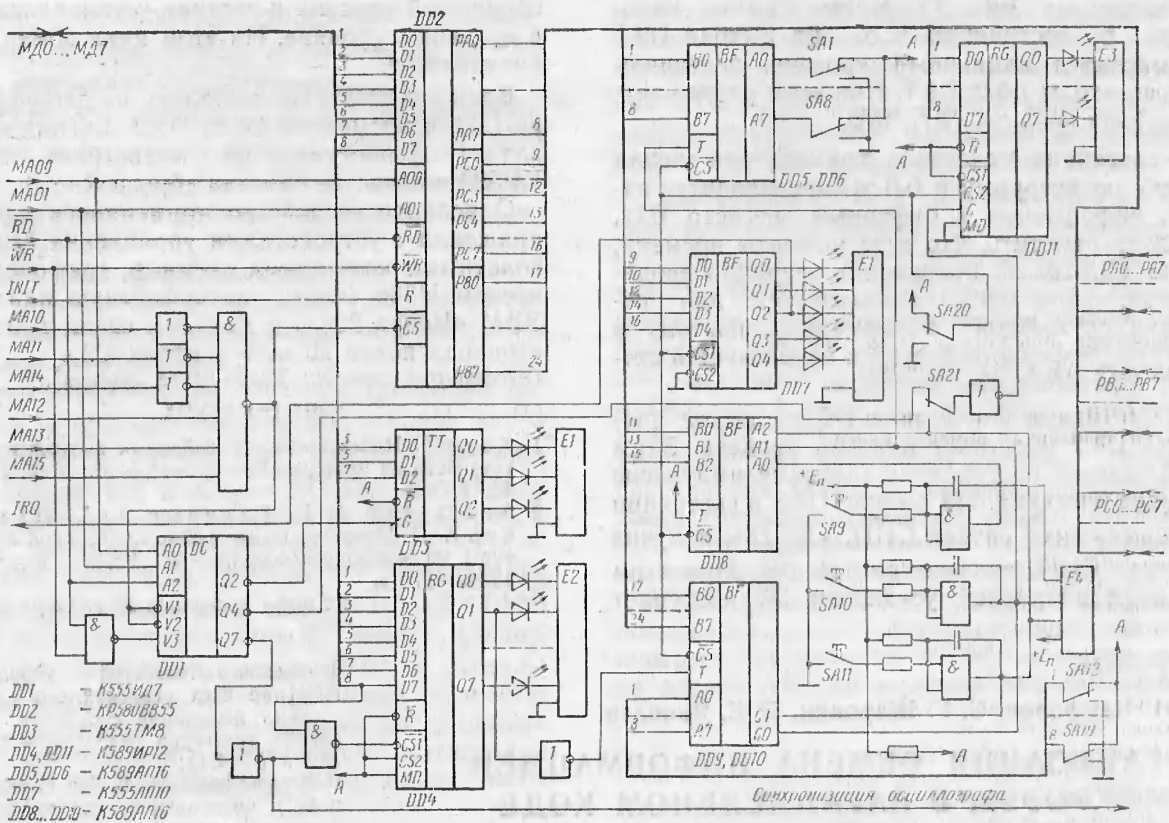


Рис. 2. Принципиальная схема параллельного интерфейса

Работа ППИ в режиме 0

Режим 0 — синхронный ввод-вывод данных по трем 8-разрядным каналам. Канал С может быть разделен в этом режиме на два 4-разрядных канала по направлению обмена. При работе ППИ в этом режиме никаких синхронизирующих сигналов от ВУ не требуется.

При выводе данных из микроЭВМ информация с МД записывается в регистр соответствующего канала и сразу же передается на его выходы. При вводе данных с ВУ информация с каналов А, В, С записывается в регистры (без стробирования) и затем по заднему фронту сигнала \overline{RD} передается на МД микроЭВМ. Временные диаграммы работы схемы в режиме 0 приведены на рис. 3. К достоинствам режима следует отнести простоту реализации обмена данными с ВУ, а также возможность сопряжения микроЭВМ с ВУ, имеющими больший формат данных, чем разрядность МД микроЭВМ. Недостаток режима 0 — при вводе данных из БИС ППИ в микроЭВМ возможно считывание ложной информации, если в процессе выполнения команды чтения происходит изменение данных на входах каналов ППИ.

Существенно, что после включения питания и подачи импульса начальной установки все каналы ППИ устанавливаются в режим ввода данных. Поэтому при программировании БИС для вывода данных из микроЭВМ после записи управляющего слова в ППИ возможна выдача ВУ случайной информации, записанной в регистре канала ранее или установившейся в нем сразу после включения питания.

Последовательность ввода-вывода информации в режиме 0*

Используя схему, приведенную на рис. 4, и программу 1 (SA20 — в положение «РАБ», SA21 — в положение «ДШ»), можно убедиться в следующем:

при работе БИС ППИ в режиме 0 передача данных со входа соответствующего канала на МД микроЭВМ осуществляется на третьем машинном цикле выполнения команды чтения данных из канала, при поступ-

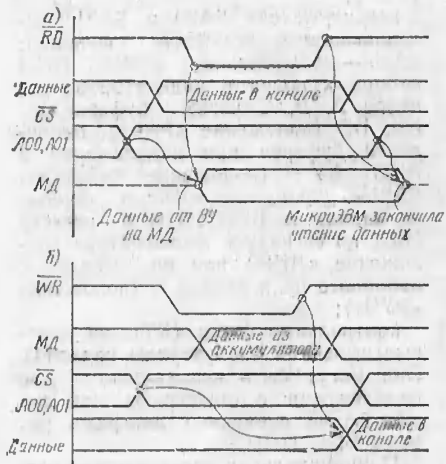


Рис. 3. Временные диаграммы работы БИС ППИ в режиме 0: ввод (а), вывод (б)

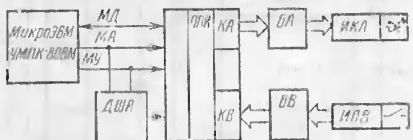


Рис. 4. Схема проверки последовательности ввода-вывода информации в режиме 0

* При проведении исследований необходимо использовать микроЭВМ с режимами выполнения программ по командам и машинным циклам, а также возможностью отображения содержимого регистров МП БИС на дисплее. Такие функциональные возможности имеют, например, учебные микроЭВМ УМПҚ-80ВМ и МП-589 [2-4].

лении импульса \overline{RD} на вход \overline{RD} БИС;

запись данных в канал происходит по окончании третьего машинного цикла выполнения команды вывода

ПРОГРАММА 1

```
0800 3E8B LOAD3: MUI A, C10 ; ЗАПИСЬ УПРАВЛЯЮЩЕГО
0802 D3B3 OUT RCUW ; СЛОВА В РЕГИСТР
0804 D8E1 LOUFD: IN KB ; ЧТЕНИЕ ДАННЫХ ИЗ КА
0806 D3B0 OUT KA ; ЗАПИСЬ ДАННЫХ В КА
0808 C340B JMF LOUFD ; ЗАМКЛИВАНИЕ
```

ПРОГРАММА ДЛЯ ИССЛЕДОВАНИЯ РАБОТЫ БИС ППИ В РЕЖИМЕ С

данных в соответствующий канал, при появлении положительного фронта импульса WR с МУ микроЭВМ.

Работа ППИ в режиме 0 при изменении направления обмена информацией

При перепрограммировании каналов А или В с ввода на вывод информации в ВУ могут выдаваться произвольные данные, имеющиеся в соответствующем канале до момента изменения направления обмена. Это можно проверить, используя запись данных в ИКА по сигналу АСК.А с НПС (SA21 — в положение «РУЧ») и клавишу STB.A для отправки буферов канала А (БА).

Особенностью работы канала С является то, что при программировании для вывода данных в интервале времени после записи управляющего слова до записи информации потенциалы его выходов соответствуют «Log.0».

Работа ППИ в режиме 1

Режим 1 — асинхронный односторонний обмен данными по каналам А и В с помощью управляющих сигналов, передаваемых по каналу С. Направление обмена для каналов А и В определяется управляющим сло-

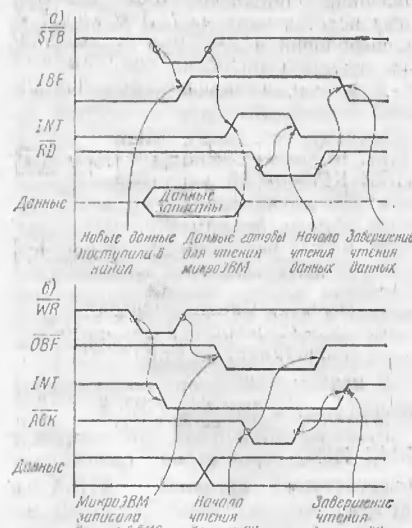


Рис. 5. Временные диаграммы работы БИС ППИ в режиме 0: ввод (а), вывод (б)

вом, записанным в регистр управления. Недействительные для управляющих сигналов разряды канала С могут быть использованы для ввода-вывода информации (рис. 5, а, б).

При работе каналов А и В на ввод данных информация записывается во входной регистр канала по заднему фронту сигнала STB (строб). Выходной сигнал IBF (ввод в буфер) устанавливается в «1» и поддерживается на протяжении всего времени от начала записи данных в канал до момента чтения их в микроЭВМ. На протяжении этого интервала ВУ запрещается формировать новый сигнал STB. Если сигнал будет подан при других данных на входах канала, то ранее записанная в канал информация теряется и заменяется новой. Если сигнал IBF и внутренний триггер разрешения прерываний (INTE) канала устанавливаются в «1», то по положительному фронту сигнала STB на выходе ППИ формируется сигнал INT. Этот сигнал может быть использован для информирования микроЭВМ о готовности данных к вводу.

При работе каналов А и В на вывод данных запись в схему осуществляется с помощью импульса записи, подаваемого на вход WR. В момент окончания записи формируется выходной управляющий сигнал OBF (вывод в буфер), указывающий ВУ, что данные готовы к выдаче. Вывод данных в ВУ осуществляется по сигналу АСК (прием), поступающему с ВУ. Для информирования микроЭВМ о готовности канала к получению новых данных может быть использован сигнал INT. На интервале вывода данных в ВУ этот сигнал установлен на низком уровне и будет устанавливаться в «1» при наличии единичных уровней сигналов АСК, OBF и соответствующего внутреннего триггера разрешения прерываний INTE.

Возможны два способа использования сигнала INT для информирования микроЭВМ о готовности канала к вводу-выводу данных: схематехнический — выход INT ППИ подключается ко входу INT МИ БИС или блока приоритетных прерываний; программный — микроЭВМ считывает слово состояния ППИ из канала С и реагирует в зависимости от содержимого разрядов INT.

Управление обменом информацией по каналам А и В в режиме 1

Разрешение или запрет обмена информацией по каналам А или В осуществляется путем записи в регистр управления ППИ слов, определяющих состояние внутренних триггеров INTE0 и INTE2. В этом можно убедиться, используя схему, приведенную на рис. 6, и программу 2

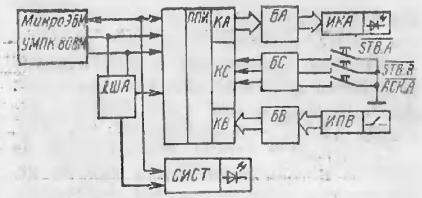


Рис. 6. Схема для исследования процесса обмена информацией по каналам А и В в режиме 1

ПРОГРАММА 2

```
0808 3EAF LOAD3: MUI A, C01 ; ЗАПИСЬ УПРАВЛЯЮЩЕГО
0810 D3B3 OUT RCUW ; СЛОВА В РЕГИСТР
0812 3E05 MUI A, SET0 ; УСТАНОВКА ТРИГГЕРА
0814 D3B3 OUT RCUW ; INTEN
0816 3E0D MUI A, SET2 ; УСТАНОВКА ТРИГГЕРА
0818 D3B3 OUT RCUW ; INTE2
0820 D8E2 WAIT1: IN KB ; ЧТЕНИЕ ДАННЫХ ИЗ КА
0822 D8E2 WAIT2: IN KB ; ЧТЕНИЕ ДАННЫХ В КА
0824 C609 ANI D0H ; СИГНАЛА INT.В
0826 C4170B JZ WAIT1 ;
0828 D8B1 IN KB ; ЧТЕНИЕ ДАННЫХ ИЗ КА
0830 C4170B JZ WAIT2 ;
0832 D8E2 WAIT2: IN KB ; ЧТЕНИЕ ДАННЫХ В КА
0834 C609 ANI D0H ; СИГНАЛА INT.А
0836 C4220B JZ WAIT2 ;
0838 C3170B JMF WAIT1 ; ЗАМКЛИВАНИЕ
```

ПРОГРАММА ИССЛЕДОВАНИЯ РАБОТЫ БИС ППИ В РЕЖИМЕ 1

(SA20 — в положение «РАВ», SA21 — в положение «ДШ»).

Использование разрядов канала С для ввода-вывода информации

При работе БИС ППИ в режиме 1 некоторые разряды канала С остаются недействительными под управляющие сигналы. Так, при назначении каналов А и В для ввода данных по режиму 1 свободными остаются разряды КС4 и КС5. Их можно использовать для ввода или вывода информации с помощью коман-

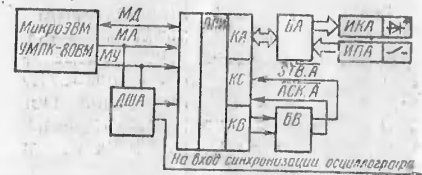


Рис. 7. Схема для исследования временных диаграмм работы БИС ППИ в режиме 1

ПРОГРАММА 3

```
082C 3E8B LOAD3: MUI A, C11 ; ЗАПИСЬ УПРАВЛЯЮЩЕГО
082E D3B3 OUT RCUW ; СЛОВА В РЕГИСТР
0830 3E09 SET1: MUI A, SET1 ; УПРАВЛЯЮЩЕГО СЛОВА
0832 D3B3 OUT RCUW ; УСТАНОВКА ТРИГГЕРА INTE1
0834 D3B3 MUI A, C3H ; ФОРМИРОВАНИЕ
0836 D3B1 OUT KA ; СИГНАЛА STB.А
0838 2F CMA ;
083B D3B1 OUT KB ;
083E 2F CMA ;
0840 D3B1 OUT KB ; ЧТЕНИЕ
0842 D3B0 IN KB ; СИГНАЛА STB.А
0844 C3300B JMF SET1 ; ЗАМКЛИВАНИЕ
```

ПРОГРАММА ДЛЯ ИССЛЕДОВАНИЯ ВРЕМЕННЫХ ДИАГРАММ РАБОТЫ БИС ППИ В РЕЖИМЕ 1 (СВОБОД)

ПРОГРАММА 4

```
0843 3EAD LOAD3: MUI A, C10 ; ЗАПИСЬ УПРАВЛЯЮЩЕГО
0845 D3B3 OUT RCUW ; СЛОВА В РЕГИСТР
0847 D3E0 SET0: MUI A, SET2 ; УСТАНОВКА ТРИГГЕРА
0849 D3B3 OUT RCUW ; INTEN
084B D3B3 MUI A, C3H ; ФОРМИРОВАНИЕ
084D D3B0 OUT KB ; СИГНАЛА АСК.А
084F D3B0 OUT KA ;
0851 2F CMA ;
0852 D3B1 OUT KB ;
0854 2F CMA ;
0855 D3B1 IN KB ; ЧТЕНИЕ АСК.А
0857 C3470B JMF SET0 ; ЗАМКЛИВАНИЕ
```

ПРОГРАММА ДЛЯ ИССЛЕДОВАНИЯ ВРЕМЕННЫХ ДИАГРАММ РАБОТЫ БИС ППИ В РЕЖИМЕ 1 (ВЫВОД)

ды чтения канала С и клавиши STB.A на НПС (соответствующей разряду КС4) или программы циклической установки (сброса) разрядов КС4 или КС5, блокируя при этом буферы каналов (SA21 — в положение «БЛК»).

С помощью схемы, приведенной на рис. 7, и программ 3 и 4 можно исследовать осциллограммы сигналов, используемых при вводе и выводе данных по каналу А в режиме 1, а также влияние состояния триггеров INTE1 и INTE2 на эти сигналы.

Работа БИС ППИ в режиме 2

Режим 2 — двунаправленный асинхронный обмен информацией по каналу А, который используется в качестве буфера двунаправленной магистрали. Сигналы управления представляют собой логическую комбинацию сигналов, используемых при работе канала А в режиме 1. Основным отличием режима 2 от режима 1 является то, что информация, записанная в регистр канала А, появляется на его выходах не сразу, а после формирования ВУ сигнала АСК.А. До этого момента выходы канала А находятся в состоянии высокого импеданса.

Управление обменом информацией в режиме 2

Используя схему, аналогичную приведенной на рис. 6 (добавив к ней НПА и сигнал STB.A), и программу 5 (переключатель SA21 — в положе-

```

ПРОГРАММА 5
085A 3E06 LOAD:MI A,SET0 ; ЗАПИСЬ УПРАВЛЯЮЩЕГО
085C D3B3 OUT RCW ; СЛОВА В РЕГИСТР
                                ; УПРАВЛЯЮЩЕГО СЛОВА
085E 3E05 MI A,SET0 ; УСТАНОВКА ТРИГГЕРА
0860 D3B3 OUT RCW ; INTE0
0862 3E09 MI A,SET1 ; УСТАНОВКА ТРИГГЕРА
0864 D3B3 OUT RCW ; INTE1
0866 3E0D MI A,SET2 ; УСТАНОВКА ТРИГГЕРА
0868 D3B3 OUT RCW ; INTE2
085A D6B2 TESTB:IN KC ; ОЖИДАНИЕ УСТАНОВКИ
085C E601 ANI 01H ; СИГНАЛА INT.B
085E CA760B JZ TESTA
0871 D6B1 IN KB ; ЧТЕНИЕ ДАННЫХ ИЗ KB
0873 C37F0B JMP WRIT ; ПЕРЕХОД К ЗАПИСИ В КА
0875 D6B2 TESTA:IN KC ; ОЖИДАНИЕ УСТАНОВКИ
0877 E620 ANI 01H ; СИГНАЛА INT.A
0879 CA6A0B JZ TESTB
087D D6B0 IN KA ; ЧТЕНИЕ ДАННЫХ ИЗ КА
087F D3B0 WRITE:OUT KA ; ЗАПИСЬ ДАННЫХ В КА
0881 D6B2 WAIT:IN KC ; ОЖИДАНИЕ СИГНАЛА
0883 E60B ANI 0BH ; СИГНАЛА INT.A
0885 CA810B JZ WAIT
0887 C36A0B JMP TESTB ; ЗАКЛИВАНИЕ
    
```

ПРОГРАММА ДЛЯ ИССЛЕДОВАНИЯ РАБОТЫ БИС ППИ В РЕЖИМЕ 2

ние «РУЧ»), можно изучить влияние управляющих сигналов STB.B, STB.A, АСК.А и триггеров INTE0... INTE2 на работу БИС ППИ в режиме 2.

Ввод-вывод информации через канал С

При работе канала А в режиме 2 и канала В в режиме 0 можно занять незадействованные под управляющие сигналы разряды КС0... КС2 канала С под ввод-вывод информации. Направление обмена определяется управляющим словом, записанным в регистр управления БИС. Убедиться в этом можно, используя команду чтения канала С, клавишу STB.B (соответствующую разряду КС2) или программу циклической установки (сброса) разрядов КС0... КС2, блокируя буферы каналов.

Ниже приведены примеры, иллю-

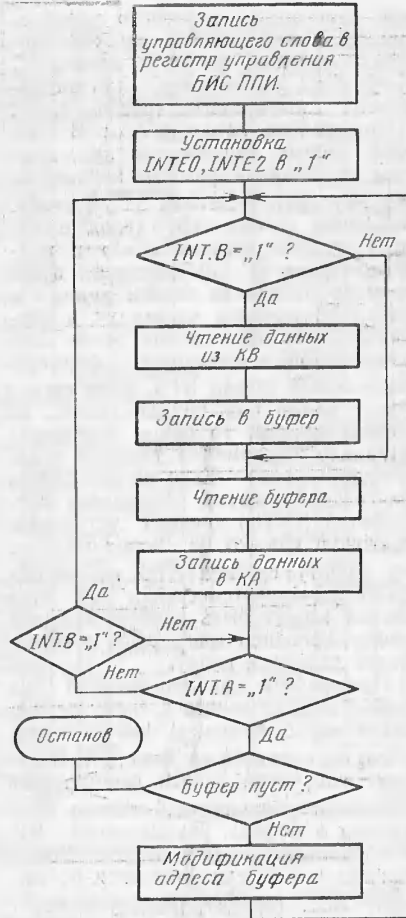


Рис. 8. Алгоритм сопряжения двух ВУ с различными скоростями ввода-вывода

```

ПРОГРАММА 6
088A 3EAF LOAD:MI A,OUT ; ЗАПИСЬ УПРАВЛЯЮЩЕГО
088C D3B3 OUT RCW ; СЛОВА В РЕГИСТР
                                ; УПРАВЛЯЮЩЕГО СЛОВА
088E 3E05 MI A,SET0 ; УСТАНОВКА ТРИГГЕРА
0891 D3B3 OUT RCW ; INTE0
0893 3E0D MI A,SET2 ; УСТАНОВКА ТРИГГЕРА
0895 D3B3 OUT RCW ; INTE2
0897 D6B2 IN KC ; ЧТЕНИЕ ДАННЫХ ИЗ KC
0899 06 NIP ;
089A 06 NIP ;
089B 01000B LXT B,ADDR ; УСТАНОВКА СЧЕТЧИКОВ
089C 11000B ENTRY:IN B,ADDR ; АДРЕСОВ БУФЕРА
089E D6B2 LOOP:IN KC ; ПРОВЕРКА СОСТОЯНИЯ
08A1 E601 ANI 01H ; СИГНАЛА INT.B
08A3 CA6C0B JZ MT ; ЕСЛИ INT.B=1,
08A5 D6B1 IN KB ; ЧТЕНИЕ ДАННЫХ ИЗ KB
08A7 D6B2 TESTB:IN KC ; ПРОВЕРКА СОСТОЯНИЯ
08A9 E602 STAX B ; ЗАПИСЬ ДАННЫХ В БУФЕР
08AB 03 INK B ; МОДИФИКАЦИЯ АДРЕСА БУФЕРА
08AC 1A MT:INX D ; ЧТЕНИЕ ДАННЫХ ИЗ БУФЕРА
08AD E6B0 OUT KA ; ЗАПИСЬ ДАННЫХ В КА
08AF D6B2 CNT:IN KC ; ПРОВЕРКА СОСТОЯНИЯ
08B1 E60B ANI 0BH ; СИГНАЛА INT.A
08B3 D6C0B JZ TEST ; ЕСЛИ INT.A=0, ПЕРЕХОД К
                                ; ПРОВЕРКЕ СИГНАЛА INT.B
08B6 21010B LXT B,ADDR ;
08B9 7E FIG:IN A,K ; ПРОВЕРКА СОСТОЯНИЯ
08BA 12 STAX D ; БУФЕРА
08BB 13 INK D ;
08BC 79 MOU A,C ;
08BD 95 SUB I ;
08BE F6B0B JZ END ; ЕСЛИ БУФЕР ПУСТ,
                                ; ВЫХОД ИЗ ПРОГРАММЫ
08C1 CA8B0B JZ DCRM ; ЕСЛИ БУФЕР НЕ ПУСТ,
                                ; МОДИФИКАЦИЯ БУФЕРА
08C4 23 INK H ;
08C5 C4900B JMP FIFO ;
08C8 06 DCRM:INX B ; УМЕНЬШЕНИЕ СЧЕТЧИКА
                                ; АДРЕСА
08C9 C3E00B JMP ENTRY ; ПЕРЕХОД К НОВОМУ
                                ; ЦИКЛУ ОБМЕНА
08CC D6B2 TEST:IN KC ; ПРОВЕРКА СОСТОЯНИЯ
08CE E601 ANI 01H ; СИГНАЛА INT.B
08D0 C3A10B JNZ LOOP1 ;
08D3 C3A70B JMP CNT ; ЕСЛИ INT.B=0, ПЕРЕХОД
                                ; К ПРОВЕРКЕ СОСТОЯНИЯ
                                ; БУФЕРА INT.A
08D6 FF DMB:RS1 Z ; ВЫХОД В МОНИТОР
    
```

ПРОГРАММА ДЛЯ СОПРЯЖЕНИЯ ДВУХ ВЕДУЩИХ УСТРОЙСТВ С РАЗЛИЧНОЙ СКОРОСТЬЮ ВВОДА-ВЫВОДА ИНФОРМАЦИИ

стрирующие некоторые способы обмена информацией с микроЭВМ в параллельном коде.

Пример 1. При создании и эксплуатации микропроцессорных систем нередко возникает необходимость сопряжения двух или нескольких ВУ с различной скоростью ввода-вывода информации (например, перфоратор ПЛ-150 и фотосчитыватель FS-1501). Для того чтобы не ограничивать по скорости ввода-вывода информации более быстродействующее из них, необходима специальная программа. Она должна обрабатывать запросы на выдачу и прием информации от ВУ и в случае, когда запросы на выдачу приходят чаще, чем на прием (т. е. устройство ввода работает быстрее, чем устройство вывода), организовать специальную буферную область памяти. В этой области должна храниться «избыточная» информация, которая по мере поступления запросов от устройства вывода выдавалась бы, а буферная область при этом модифицировалась (т. е. она должна представлять собой очередь типа FIFO*). Алгоритм, приведенный на рис. 8, и программа 6 являются иллюстрацией способа сопряжения «медленного» устройства вывода (ИКА) и «быстрого» устройства ввода (НВ). Чередуя подачу сигналов STB.B и АСК.А произвольным образом, можно убедиться, что в любом случае при i-м нажатии на клавишу АСК.А на ИКА отображается информация, введенная при i-м нажатии клавиши STB.B.

Пример 2. В системе из двух микроЭВМ часто одна является главной (ведущей), другая — подчиненной (ведомой). Обмен данными организуется по командам ведущей. Такое построение экономит время и увеличивает производительность системы, поскольку обработка массивов данных ведется параллельно (с ведомой) с выполнением основной программы (в ведущей).

Для сопряжения в режиме «ведущей — ведомой» двух микроЭВМ, построенных на базе МПК КР580 (рис. 9), может использоваться БИС ППИ КР580ВВ55, запрограммированная для двунаправленного асинхронного обмена (режим 2). При этом БИС может рассматриваться как устройство полудуплексной параллельной передачи данных — информация передается либо от ведущей к ведомой, либо наоборот, но не в обоих направлениях одновременно.

В качестве сигналов ввода-вывода используются ИВ.А и ОВ.А.

Ведомая микроЭВМ связывается с БИС ППИ через схему сопряжения посредством сигналов STB.A и АСК.А. Импульс STB.A записывает информацию с МД ведомой микро-

* FIFO (First Input, First Output) — первый вошел, первый вышел.

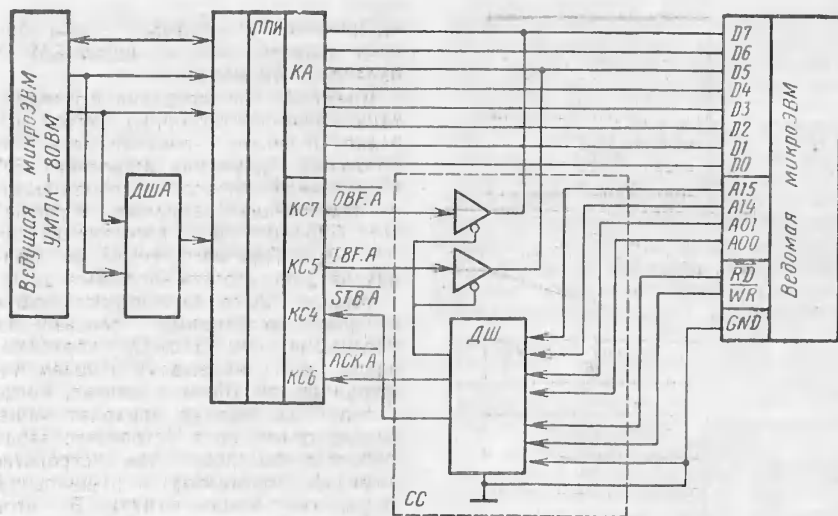


Рис. 9. Функциональная схема устройства сопряжения двух микроЭВМ

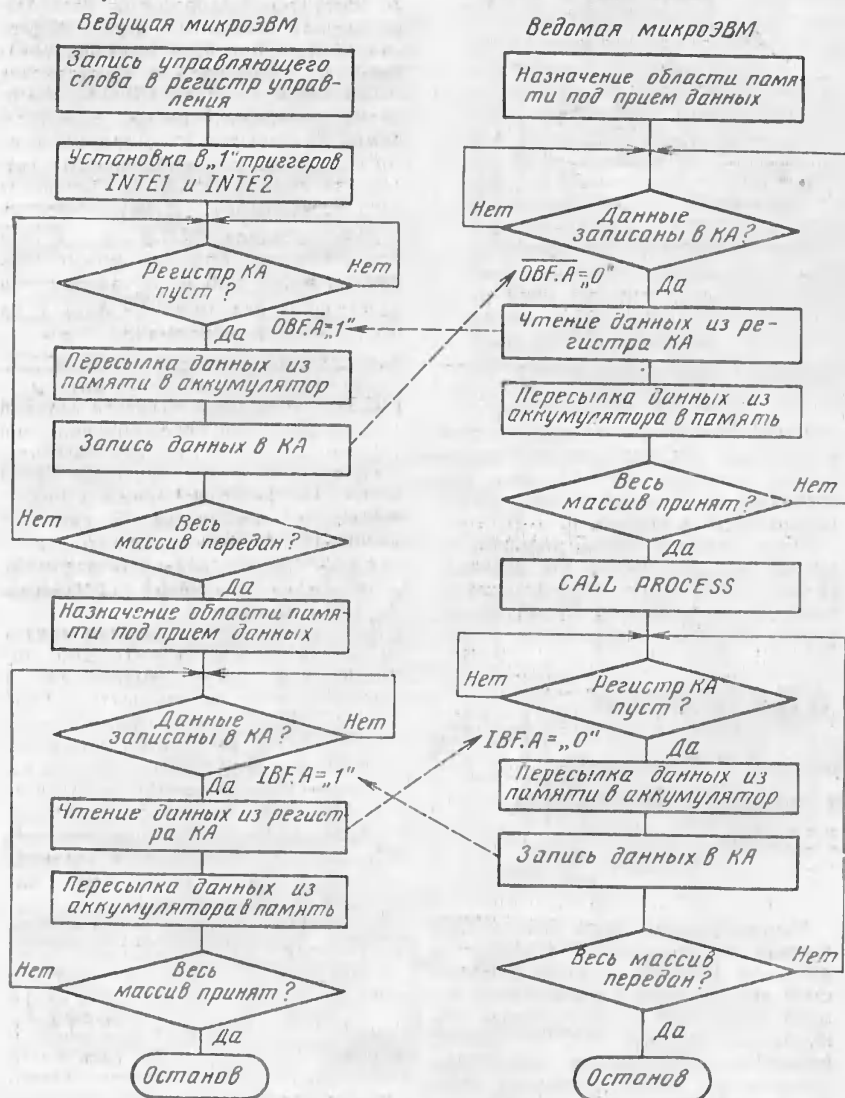


Рис. 10. Алгоритмы обмена массивами данных в двухмашинной системе

ЭВМ в канал А ППИ и одновременно устанавливает сигнал $\overline{IBF.A}$. Импульс $\overline{ASK.A}$ разрешает выдачу информации из канала А и устанавливает сигнал $\overline{OBF.A}$. Существенно, что в ведомую микроЭВМ сигналы $\overline{IBF.A}$ и $\overline{OBF.A}$ поступают в инвертированном виде.

При передаче массива данных из ведущей в ведомую каждая микроЭВМ проверяет только состояние сигнала $\overline{OBF.A}$. После обработки в ведомой массив может быть передан назад, в ведущую. При этом проверяется только сигнал $\overline{IBF.A}$ (рис. 10).

В начале цикла вывода ведущая микроЭВМ проверяет наличие информации в регистре канала А. Если он пуст, выполняется команда записи, при этом данные из аккумулятора переписываются в регистр КА и сигнал $\overline{OBF.A}$ устанавливается в «0». В это время ведомая микроЭВМ использует чтение канала С для определения состояния сигнала $\overline{OBF.A}$. Если $\overline{OBF.A}$ установлен в «0», она считывает данные из канала А в аккумулятор. Таким образом, весь массив передается в ведомую микроЭВМ. Она выполняет его обработку (в программе PROCESS) и начинает проверять состояние сигнала $\overline{IBF.A}$, определяя тем самым наличие информации в регистре КА, не считанной ведущей микроЭВМ. Если регистр КА пуст, то ведомая микроЭВМ записывает в регистр данные из аккумулятора, устанавливая сигнал $\overline{IBF.A}$ в «1». В это время ведущая микроЭВМ, при возникновении необходимости в приеме обработанного массива, начинает проверять состояние сигнала $\overline{IBF.A}$ и при его установке в «1» считывает данные из КА. При этом $\overline{IBF.A}$ сбрасывается в «0», система готова к новому циклу обмена. Таким образом, весь массив передается из ведомой микроЭВМ в ведущую.

Адрес для справок:

193498, Москва, МИЭТ

ЛИТЕРАТУРА

1. Алексенко А. Г., Галицын А. А., Иванников Л. Д. Просктирование радиоэлектронной аппаратуры на микропроцессорах.— М.: Радио и связь, 1984.— 270 с.
2. Горбунов В. Л., Панфилов Д. И., Преснухин Д. Л. Микропроцессоры. Основы построения микроЭВМ.— М.: Высшая школа, 1984.— 144 с.
3. Красавин В. Н., Панфилов Д. И., Романенко О. А., Шаронин С. Г. Обучающая система на базе микропроцессорного комплекса серии КР580.— Электронная промышленность, 1983, № 9, с. 38.
4. Горбунов В. Л., Панфилов Д. И. Микропроцессоры. Лабораторный практикум.— М.: Высшая школа, 1984.— 140 с.

Статья поступила 24 октября 1985 г.

А. П. Дмитренко, Е. Ф. Накалов

КОНТРОЛЛЕР КЛАВИШНОГО ПУЛЬТА

Предлагаемый контроллер клавишного пульта (рис. 1) обеспечивает функционирование 64 клавиш и ориентирован на совместную работу с программируемым параллельным интерфейсом (ППИ) КР580ВВ55.

А2 счетчика, проходя через дешифратор D6, сканируют клавиатуру по строкам, а сигналы с выходов А3, А4, А5, поступаая на мультиплексор D7, производят опрос столбцов клавиатуры. Таким образом, каждой

пает на схему D3.2, разрешая перезапуск одновибратора по входу D импульсами генератора.

С инверсного выхода одновибратора сигнал поступает на схему D3.1 и запрещает прохождение тактовых импульсов на счетчик, прекращая сканирование клавиатуры; при этом на выходах А0...А5 счетчика фиксируется код клавиши. Таким образом, дребезг контактов клавиши не влияет на установившийся код.

Сигнал СТР стробирует запись содержимого разрядов А0...А5 во входной регистр ППИ. Ответным сигналом со стороны ППИ является сигнал подтверждения приема (ППР). Сигнал ППР, пройдя через инвертор, поступает на схему D3.1 и запрещает сканирование клавиатуры до тех пор, пока микропроцессор не прочтает содержимое входного регистра ППИ.

Дребезг, возникающий при отпущении клавиши, также перезапускает одновибратор, и когда на его входе установится «Лог 1», работа одновибратора прекращается. Длительность импульса τ (рис. 2), вырабатываемого одновибратором, должна быть больше периода дребезга контактов.



Рис. 2. Временные диаграммы работы клавишного пульта

При незначительном изменении схема контроллера клавиатуры может быть расширена до большего количества клавиш.

Адрес для справок: 290005, г. Львов ул. Ватутина, 5, Политехнический институт. Статья поступила 24 октября 1985 г.

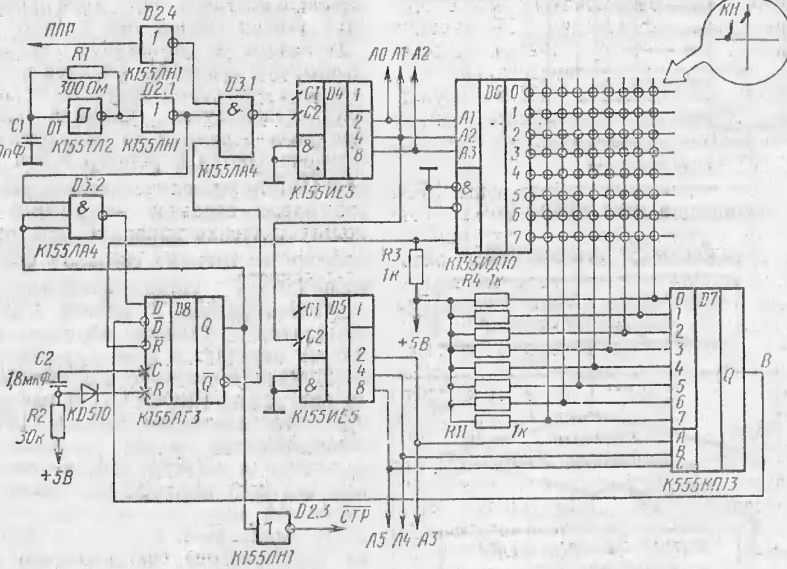


Рис. 1. Принципиальная схема контроллера клавишного пульта

Функционально схема состоит из следующих узлов: генератора тактовых импульсов (D1, D2.1), узла сканирования клавиатуры (D4, D5, D6, D7), узла подавления дребезга клавиш (D8, D3.2).

Тактовые импульсы, пройдя через схему разрешения D3.1, тактируют двончный счетчик (D4, D5). Сигнала с выходов младших разрядов А0, А1,

нажатой клавише соответствует свой конкретный код, определенный содержимым разрядов А0...А5. Этот код поступает на входной регистр ППИ, работающего в режиме 1.

После нажатия любой клавиши на выходе мультиплексора D7 возникает спадающий фронт, запускающий по входу D одновибратор D8. Импульс, формируемый одновибратором, посту-

Е. М. Злотник, И. К. Стежко, В. В. Анищенко, С. И. Киркорев

«ЭЛЕКТРОННЫЙ ДИСК» ДЛЯ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ НА БАЗЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Вычислительные комплексы типа ДВК и ВУМС [1] базируются на микроЭВМ с архитектурой ЭВМ «Электроника 60» (рис. 1).

Операционная система (ОС) этих комплексов включает в себя управляющую систему, систему программирования, редактирования и отладки программ, файловую и сервисную системы.

КВАЗИДИСК

Исполнительная часть ОС (ИОС) состоит из резидентной (РИОС) и дисковой (ДИОС). РИОС полностью размещается в оперативной памяти микроЭВМ и обеспечивает минимальные функции ИОС [2]. ДИОС размещается на гибком магнитном диске и при вызове отдельных программ загружается в оперативную память микроЭВМ, причем произво-

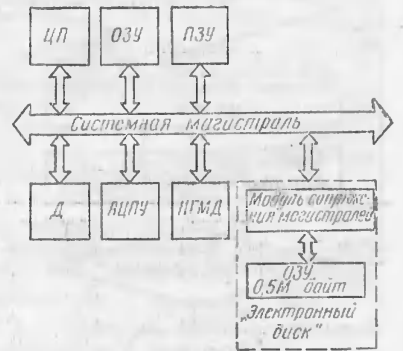


Рис. 1. Архитектура вычислительных комплексов типа ДВК и ВУМС

длительность системы заметно снижается из-за задержек, связанных с обращением к магнитному диску. Увеличить вычислительную мощность можно путем хранения всей или части ДИОС во вспомогательном ОЗУ большого объема.

Разработанный «электронный диск» реализует функции вспомогательного ОЗУ большого объема (см. рис. 1). В состав устройства входят модуль сопряжения магистралей (МСМ) и ОЗУ (0,5М байт) на платах промышленного исполнения (ПЗ, П5).

МикроЭВМ и «электронный диск» взаимодействуют по принципу «окна» на системной магистрали. «Окно» размещается в области адресующих регистров, его адресное пространство (512 байт) начинается с адреса 165000. Обмен информацией ведется блоками (стандартными единицами обмена в ОС емкостью 512 байт) или 16-разрядными словами (группой слов).

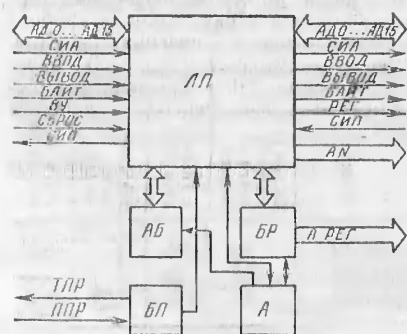


Рис. 2. Функциональная схема модуля сопряжения магистралей (МСМ)

Функциональная схема модуля сопряжения магистралей показана на рис. 2. Формируемый ЦП 16-разрядный адрес позволяет адресоваться к 64К байт. Увеличение адресного пространства микроЭВМ требует увеличения разрядности адреса. Адресный блок (АБ), состоящий из регистра номера блока и регистра номера слова в блоке, формирует N-разрядный адрес (N—число разрядов, необходимых для обращения к вспомогательному ОЗУ). ОЗУ реализовано на микросхемах ЗУ динамического типа, требующих регенерации не реже, чем через каждые 2 мс. В состав блока регенерации (БР) входят таймер регенерации, счетчик регенерации и адресные передатчики.

Логика протокола (ЛП) служит для организации интерфейса между микроЭВМ и стандартными платами оперативной памяти.

При совместной работе микроЭВМ и «электронного диска» может возникнуть конфликтная ситуация — микроЭВМ обращается к ячейке вспомогательной памяти, которая в данный момент регенерируется. Для правильного функционирования системы применяется схема арбитража (А). Функциональная схема арбитража (рис. 3) включает в себя триггеры: запроса обращения ТЗО (D1.1) к «электронному диску», цикла обращения ТЦО (D1.2), запроса регенерации памяти ТЗР (D2.1), цикла регенерации ТЦР (D2.2), цикла памяти ТЦП (D8.1) и вспомогательную логику (D3, D4, D5.1, D6.1, D7.1). Если вспомогательная оперативная память свободна, то запрос от генератора синхросигнала регенерации обрабатывается цепочкой триггеров

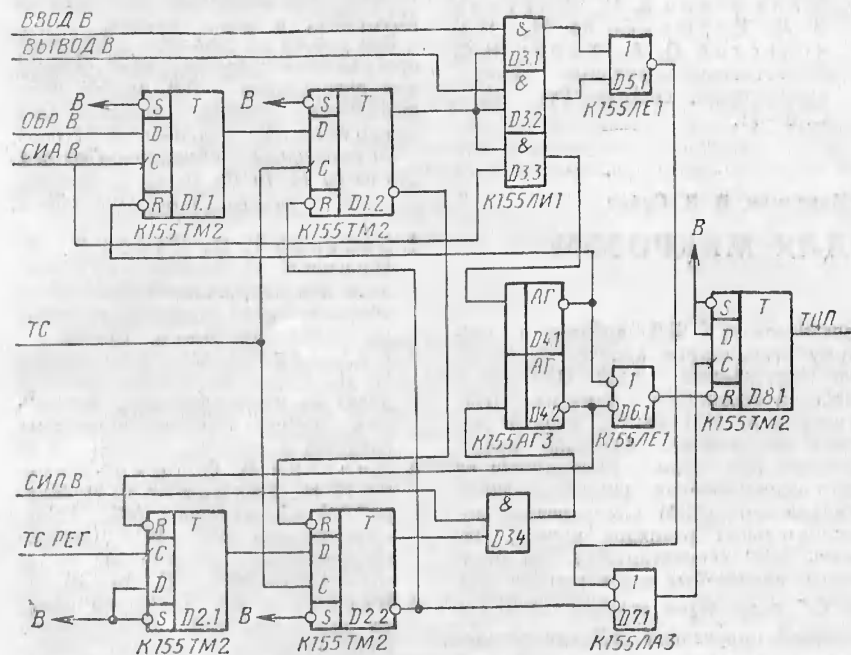


Рис. 3. Функциональная схема арбитража

ТЗР—ТЦР—ТЦП, а запрос от центрального процессора—цепочкой ТЗО—ТЦО—ТЦП.

Рассмотрим работу цепочки ТЗР—ТЦР—ТЦП. Тактовые импульсы регенерации (ТС РЕГ) с периодом 30 мкс (2 мс/64) поступают на вход синхронизации триггера ТЗР. Состояние этого триггера переписывается тактовым сигналом ТС с частотой 4,608 МГц в триггер ТЦР, устанавливающий триггер ТЦП. Установка ТЦП инициализирует цикл обращения к вспомогательной памяти. При появлении сигнала СИП В (вырабатывается при окончании цикла памяти) триггеры ТЗР и ТЦП сбрасываются интегральной схемой формирования импульса D4.2. При появлении сигнала ОБР В цепочка триггеров ТЗО—ТЦО—ТЦП работает аналогично.

Если один из запросов поступает в тот момент, когда обработка другого еще не завершена, то до окончания данного цикла памяти обслуживание более позднего запроса задерживается.

Блок прерываний (БП) синхронизирует работу устройства в вычислительной системе.

Работоспособность «электронного диска» в операционной среде поддерживается драйвером устройства (рис. 4).

Драйвер программируется на языке макроассемблера в позиционно

```

TITLE DZ
IDENT Y84.00
MACRO .GETDF, .DRBEC, .DRENZ, .DRAST, .DRFIN
GETDF
*IF NDF MMC#T, MMC#T=0
*IF NDF ERL#G, ERL#G=0
*IF NDF #TIM#IT, TIM#IT=0
*IF NDF DZ#VEC, DZ#VES=350
*IF NDF DZ#CSR, DZ#CSR=16.000
DZDSIZ=756
DZSTS=276
BUFF#Q.BUFF-Q.BEKA
WCNT#Q.WCNT-Q.BEKA
.DRBEC DZ,DZ#VEC,DZSIZ,DZSTS
MOV DZCDE,R5
MOV @R5,R4
BTS #100,R4
MOV R4,@#DZ#CSR
PDS
.DRAST DZ,Q
MOV #DZDSIZ-100,R4
MOV DZCDE,R5
MOV WCNT(R5),(PC)+
PI: .WORD 0
BPL KB
MFC WCNT(R5)
KB: TST FL
BIO DZEX
KI: TST WCNT(R5)
BMT WRIT
MOV (R4),@BUFF(R5)
BR CONT
WRIT: MOV @BUFF(R5),(R4)+
CONT: ADD #2,BUFF(R5)
DEC FL
BIO DZEX
CMP #DZ#CSR,R4
BNE X1
INC @#DZ#CSR
PDS
DZEX: MOV #0,@DZ#CSR
.DRFIN DZ
.DRENZ DZ
.END

```

Рис. 4. Исходный текст драйвера «Электронного диска»

независимом коде. Программа драйвера транслируется, как обычно. Затем надо переименовать оттранслированный файл в имя DD.SYS. Готовый драйвер записывается на системный диск. Для ОС ДВК версии 4 и старше информация об устройстве заносится в таблицы резидентного монитора (находящиеся в оперативной памяти) автоматически при загрузке системы. Для ранних версий

Сравнительная характеристика устройства «Электронный диск» и НГМД «Электроника 7012»

| Характеристика | Гибкий магнитный диск | Электронный диск* |
|--|-----------------------|-------------------|
| Время доступа к блоку информации 512 байт в ОС, мс | 250 | 3 |
| Скорость передачи данных, К байт/с | 30 | 150 |
| Время инициализации с проверкой на дефектные блоки, с | 43 | 8 |
| Время копирования гибкого магнитного диска, с | 137 | 52 |
| Время трансляции программы размером 10 К байт на языке макроассемблер, с | 37 | 16 |

ОС ДВК, ФОБОС, РАФОС необходимо использовать команду монитора INSTALL.

«Электронный диск» применялся при разработке системного ПО (см.

УДК 681.32

Ю. Ю. Сорокин, В. В. Лаврентьев, С. П. Максымяк, В. В. Субач

«ЭЛЕКТРОННЫЙ ДИСК» ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60М»

«Электронный квазидиск» [1] позволяет существенно уменьшить нагрузку на НГМД. При этом функции НГМД сводятся к архивизации данных, а их обработка средствами ОС производится на квазидиске. Это решение обеспечивает высокую скорость обмена данными при приемлемой емкости устройства и его невысокой стоимости.

Такое устройство можно реализовать на интегральных схемах (ИС) ЦМД-структуры [2] или путем наращивания ОЗУ для организации в нем логического диска (ЛД) [3]. Пользователи микроЭВМ «Электроника 60» или ДВК-2 уже сегодня могут

таблицу). Из таблицы видно, что «электронный диск» наряду со стандартным набором периферийных устройств увеличивает производительность вычислительной системы, т. е. снижает трудоемкость разработки программ, особенно отладки, редактирования, получения копий и др.

Модуль сопряжения магистралей конструктивно выполнен на плате размерами 280×240 мм [3] и устанавливается в субблоке микроЭВМ. Оперативная память собрана в дополнительном субблоке, связанном с МСМ.

«Электронный диск» позволяет в некоторых случаях обходиться без дорогостоящей внешней памяти произвольного доступа. Такие вычислительные комплексы реализуются двумя способами:

- ПЗУ используется для хранения операционной системы;
- «электронный диск» предварительно загружается с помощью аппаратных средств (например, перфопвода, кассетного магнитофона).

Адрес для справок: 220605, Минск, ул. Сурганова, 6, ИТК АН БССР. Телефон — 39-59-85 (Минск).

ЛИТЕРАТУРА

1. Попов А. А., Хохлов М. М., Глухман В. Л. Диалоговые вычислительные комплексы «Электроника НЦ-80-20»//Микропроцессорные средства и системы.— 1984.— № 4.— С. 61—64.
2. Игнатьев М. Б., Кибиткин В. В., Осовецкий Л. Г. Структура и особенности программного обеспечения микроЭВМ//Там же.— С. 44—47.
3. Малашев Б. М., Дшхунья В. Л., Борщенко Ю. И., Никольский О. А., Хорин В. С. Магистрально-модульные микропроцессорные системы//Там же.— С. 3—11.

организовать ЛД, добавив в систему необходимое количество серийно выпускаемых плат ОЗУ ПЗ и простой «дискетчер памяти» (ДП) (например, [4]) (8 ИС низкой степени интеграции). Адресное пространство при этом расширяется за счет использования резервных линий канала микроЭВМ для передачи дополнительных разрядов адреса. Память ЭВМ сегментируется на страницы, адресуемые через регистр ДП. В ОС включается драйвер, обслуживающий полученный ЛД как псевдоустройство со стандартной файловой структурой.

Конкретная реализация описанного устройства выполнена на базе микроЭВМ «Электроника 60» (ВУМС). Установка шести дополнительных плат ПЗ обеспечила емкость ЛД 270 блоков. Минимальная модернизация процессора М2, устройства И4 (без установки дополнительных элементов) и плат ПЗ (одна дополнительная ИС низкой степени интеграции на плату) позволила увеличить емкость ЛД до 383 блоков.

Интенсивная эксплуатация системы в течение длительного времени с ОС РАФОС показала ее высокую эффективность. «Время жизни» дискет и магнитных головок НГМД-7012 увеличилось в несколько десятков раз. Время трансляции программ уменьшилось в 8...10 раз (см. таблицу). Использование ЛД для раз-

Таблица

Затраты времени на трансляцию программ до уровня загрузочного кода

| Характеристики программ | Время трансляции программ, мин | | | | |
|-------------------------|--------------------------------|--------------------------------|-----------------|--------------------|------------------|
| | Язык | Объем исходного текста, блоков | с НГМД (без ЛД) | ЛД наз-начен на ДК | ЛД в качестве SV |
| PASCAL | | 30 | 35 | 4 | — |
| MACRO | | 110 | 25 | 3 | — |
| MACRO | | 50 | 11 | 1,2 | 0,6 |

мещения и загрузки ОС позволило без обращения к НГМД эксплуатировать минимальную версию РАФОС для подготовки программ на MACRO-11 и языке Бейсик.

Вся работа по созданию аппаратно-программных средств, необходимых для организации ЛД, может быть выполнена силами пользователей микроЭВМ.

За справками обращаться по тел. 246-05-80 (с 10 до 16 ч), г. Москва.

ЛИТЕРАТУРА

1. Зеленко Г. В., Панов В. В., Попов С. Н. Электронный квазидиск для персональной ЭВМ//Микропроцессорные средства и системы.— 1984.— № 4.— С. 79—82.
2. Кузнецов С. О., Прохоров Н. Л., Раев В. К. «Электронный диск» на цилиндрических магнитных доменах//Микропроцессорные средства и системы.— 1985.— № 4.
3. Лопатин В. С., Пархоменко П. И., Токмаков В. И. МикроЭВМ «Электроника МС 1211», «Электроника МС 1212»//Микропроцессорные средства и системы.— 1985.— № 2.— С. 14—15.
4. Черняк А. Ю. Дискетчер памяти для микроЭВМ «Электроника 60М»//Приборы и техника эксперимента.— 1985.— № 3.— С. 98—99.

Статья поступила 5 мая 1986 г.

УДК 681.322

Быстродействующая векторная вычислительная система ПС-320 // Микропроцессорные средства и системы.—1986.— № 5.— С. 3.

Рассматриваются структура, архитектура и принцип функционирования многопроцессорной векторной вычислительной системы ПС-320. В этой системе, помимо универсальных скалярных команд, предусмотрены команды синхронизации и планирования, векторные команды обработки одно- и двумерных массивов данных.

Высокая производительность до 1 млн. операций с плавающей точкой в секунду и ОЗУ емкостью до 8М байт позволяют строить на базе ПС-320 мощные персональные профессиональные ЭВМ и рабочие станции.

УДК 681.324

Шкамарда А. Н. Шестнадцатиразрядные микроЭВМ семейства SM1800 // Микропроцессорные средства и системы.—1986.— № 5.— С. 6.

Дано описание двух новых 16-разрядных моделей семейства SM1800: SM1810 и SM1814. Приводятся краткие технические характеристики, назначение, состав основных модулей и периферийных устройств, а также программного обеспечения; описываются способы поставки микроЭВМ; перспектива развития семейства SM1800.

УДК 681.326

Долкарт В. М., Степанов В. Н., Крамфус И. Р. Одноплата микроЭВМ ПМВ02 на базе МПК БИС K1810 // Микропроцессорные средства и системы.—1986.— № 5.— С. 10.

Даны аппаратные и программные характеристики высокопроизводительной одноплаты микроЭВМ на базе МПК БИС K1810, разработанной для магистрально-модульных микросредств управляющей вычислительной техники серии В9 (МСУВТ В9).

УДК 681.326+681.325.5

Говорун В. Н., Горбунов Н. В., Мамаков П. В., Рыбаков В. Г., Сытин А. Н., Холоденко Г. М. МикроЭВМ и управляющие модули на базе МПК БИС серии K1810.— Микропроцессорные средства и системы.—1986.— № 5.— С. 13.

Описаны одноплата микроЭВМ на базе микропроцессора K1810VM86, а также дополнительный контроллер и драйвер ветви.

УДК 681.322.068

Корчак А. Е. Бейсик /Fs — язык программирования школьных и бытовых компьютеров // Микропроцессорные средства и системы.—1986.— № 5.— С. 31.

Предложен проект единого языка программирования для широкого класса школьных и бытовых компьютеров. Описываются аппаратно-независимые графические и звуковые средства языка Бейсик/Fs.

УДК 681.32

Найденков А. В., Романенков В. А. Программатор ППЗУ на базе микроЭВМ «Электроника К1-20».— Микропроцессорные средства и системы.—1986.— № 5.— С. 34.

Рассматриваются принципиальная схема, программное обеспечение и технические характеристики программатора на базе микроЭВМ «Электроника К1-20».

UDC 681.322

PC-320 High-Speed Vector Computer System // Microprocessor Devices and Systems. 1986. N 5. P. 3.

The paper discusses the architecture and functions of multiprocessor vector computer system PC-320. Besides standard scalar instructions PC-320 provides manipulation on vectors, synchronization and planning instructions.

The operation speed of upto 1M floating point operations per second and 8M of RAM makes it possible to build powerful professional workstations on the basis of PC-320.

UDC 681.324

Shkamarda A. N. The New Models of SM1800 Microcomputers // Microprocessor Devices and Systems. 1986. N 5. P. 6.

The author describes two new 16-bit models of SM1800, SM1810 and SM1814 microcomputer series. The paper gives technical characteristics, the description of main modules, peripheral devices and software. The plans for development of SM1800 family is discussed.

UDC 681.326

Dolkart V. M., Stepanov V. N., Kramfus I. R. PMV02 Singleboard Microcomputer on the Basis of K1810 IC. Microprocessor Devices and Systems. 1986. N 5. P. 10.

The paper presents hardware and software features of singleboard microcomputer on the basis of K1810 IC. The microcomputer is intended for modular control systems of V9 series (MSUVT V9).

UDC 681.326+681.325.5

Govorun V. N., Gorbunov N. V., Mamaikov P. V., Rybakov V. G., Sytin A. N., Kholodenko G. M. Microcomputer and Control Modules on the Basis of K1810 IC Series // Microprocessor Devices and Systems. 1986. N 5. P. 13.

The paper describes singleboard microcomputer on the basis of KH1810VM86 microprocessor and auxiliary controller.

UDC 681.322.068

Korchak A. E. Basic/Fs — Programming Language for School and Home Computers // Microprocessor Devices and Systems. 1986. N 5. P. 31.

The author presents the project of BASIC for general use in the school and home computers. The paper describes the means for programming graphics and sound effects.

UDC 681.32

Naidenkov A. V., Romanenkov V. A. PROM Programmer on the Basis of «Electronica K1-20» Microcomputer.— Microprocessor Devices and Systems, 1986, N 5, P. 34.

Programmer diagram, software and technical characteristics based on the «Electronica K1-20» are considered in the paper.

Баран Е. Д., Кошелева Е. И., Салмина Е. М. Анализатор микропроцессорных систем // Микропроцессорные средства и системы.—1986.— № 5.— С. 44.

Даются аппаратные и программные характеристики логического анализатора АЛ-6, предназначенного для анализа логических состояний и информационных потоков в микропроцессорных системах с интерфейсом типа «Общая шина» и протоколом обмена микроЭВМ «Электроника С5-21М».

УДК 681.325

Щелкунов Н. Н., Дианов А. П. Универсальный одноплатный микроконтроллер // Микропроцессорные средства и системы.—1986.— № 5.— С. 65.

Предложена схема одноплатного микроконтроллера, построенного на микросхемах серии КР580. Микроконтроллер представляет собой функционально завершенную расширяемую микросистему с универсальным набором аппаратных средств, обеспечивающих широкое применение в системах управления реального времени.

УДК 681.326

Книгин С. Б. Многоголосый электромузыкальный синтезатор // Микропроцессорные средства и системы.—1986.— № 5.— С. 77.

Описывается цифровой многоголосый электронный музыкальный синтезатор на базе микропроцессорного комплекта КР580. Кратко рассматриваются принципы синтеза звука. Приведены структурные, принципиальные схемы и программное обеспечение.

УДК 681.335.2

Дубровский И. И., Финякин Л. Н., Кафоров В. В. Устройство параллельного ввода аналоговой информации в микроЭВМ // Микропроцессорные средства и системы. 1986, № 5, С. 81.

Изложены основные технические и программные решения, позволившие организовать одновременный ввод информации от группы (до восьми) первичных преобразователей, датчиков в микроЭВМ «Электроника Д3-28» по шине Ввод. В качестве датчиков использованы резистивные элементы, входящие в состав внешних времязадающих цепочек одновибраторов, формирующих широтно-импульсные сигналы (ШИС), эквивалентные контролируемым физическим величинам.

УДК 681.326

Васильев С. И., Леонов В. Б. Сопряжение микроЭВМ «Электроника Д3-28» с МПК БИС серии КР580.— Микропроцессорные средства и системы, 1986, № 5, с. 83.

Описано устройство преобразования управляющих и информационных сигналов интерфейса ввода-вывода микроЭВМ «Электроника Д3-28» в сигналы интерфейса микросхем серии КР580. Устройство обеспечивает скорость передачи информации до 200 К байт/с и позволяет подключить до 50 периферийных модулей.

УДК 681.325.5

Дмитренко А. П., Накалов Е. Ф. Контроллер клавишного пульта // Микропроцессорные средства и системы.—1986.— № 5.— С. 90.

Дается принципиальная схема контроллера клавишного пульта с устранением дребезга контактов клавиш.

Baran E. D., Kosheleva E. I., Salmina E. M. Microprocessor Systems Analyzer // Microprocessor Devices and Systems. 1986, N 5, P. 44.

The paper presents hardware and software features of logical analyzer AL-6 intended for analyzing logical positions and information flows in microprocessor systems with the "Common Bus" interface and Electronica S5-21M protocol.

УДК 681.325

Shchelkunov N. N., Dianov A. P. Universal Singleboard Microcontroller // Microprocessor Devices and Systems, 1986, N 5, P. 65.

The authors present the scheme of singleboard microcontroller based on KR580 IC series. It is a special functionally complete expandable microsystem with the standard set of hardware for real-time control systems.

УДК 681.326

Knigin S. B. Multivoice Electronic Musical Synthesizer // Microprocessor Devices and Systems, 1986, N 5, P. 77.

The author describes digital multivoice electronic synthesizer on the basis of KR580 microprocessor set. The principles of sound synthesis are discussed. The logical diagrams and software are given.

УДК 681.335.2

Dubrovsky I. I., Finjakin L. N., Kafarov V. V. Parallel Analog Information Input for Microcomputer.— Microprocessor Devices and Systems, 1986, N 5, p. 81.

The paper presents the basic description of hardware and software for simultaneous input of analog information from the set of primary sensor into "Electronica D3-28" microcomputer. Simple electronic devices with resistive elements and time modulation serve for the digital coding of analog input signals.

UDC

Vasiljev S. I., Leonov V. B. "Electronica D3-28" Microcomputer with KR580 Interface.— Microprocessor Devices and Systems, 1986, N 5, p. 83.

The paper describes the device for transformation of control and information signals of "Electronica D3-28" system bus into KR580 interface signals. The information transferring rate is 200 KB/sec. Up to 50 peripheral devices can be connected to the system bus.

УДК 681.32

Dmitrenko A. P., Nakalov E. F. Keyboard Controller // Microprocessor Devices and Systems.—1986.— N 5.— P. 90.

The keyboard controller is described allowing for diminishing the key press trembling noise.

ВНИМАНИЕ! ОШИБКИ ПЕРЕВОДА И ОРИГИНАЛА

в книге «Персональный компьютер фирмы ИБМ»

Издательство «Мир» выпускает много интересных книг по ЭВМ и программированию. Однако не обходится, как в любой работе, без «проколов». Мы получили письмо об этом специалиста по языку Бейсик Ю. Л. Кеткова. Письмо комментирует с. н. с. ВЦ АН СССР Г. В. Сенин, уже ряд лет плодотворно работающий на языке Бейсик с ПЭВМ серии ИБМ РС.

В начале 1986 г. издательство «Мир» опубликовало одну из первых книг по персональным ЭВМ—Р. Трейстер «Персональный компьютер фирмы ИБМ». Материал долгожданный, ведь в ближайшем будущем нам предстоит осваивать аналогичные ПЭВМ отечественного производства и производства социалистических стран. Полиграфическое качество издания выше всяческих похвал.

Однако серьезные возражения вызывает содержание 7-й главы — «Бейсик персонального компьютера фирмы ИБМ». Начнем со стилистических издержек перевода. Какую информацию читатель может извлечь, например, из таких фраз? — «При интенсивном программировании, чтобы получить правильный (?) номер следующей строки, часто бывает необходимо постоянно ссылаться (?) на номер предшествующей строки» (с. 93). Или «...это позволяет оператору переменные, введенные в одну программу, легко составлять (?) в другую, что резко сокращает время (?) выполнения программы» (с. 94). Что можно понять из описания оператора LINE INPUT? — «Иногда за этим выражением (?) следует метка (?), представляющая собой строчную константу, которая высвечивается на экране перед началом ввода» (с. 98).

Кроме того, вызывает недоумение ряд утверждений автора. Например, оператор присваивания LET «...присваивает определенное значение выражению (?)». И далее — он «является дополнительным (?)», поэтому «в действительности нет необходимости

его (?) использовать, поскольку он замедляет программирование (?)». Очень странно слышать, что оператор GOTO аналогичен оператору GOSUB с той лишь разницей, что переход здесь является безусловным (?).

При описывании ряда функций (LEFT \square ; LOG, MID, \square RIGHT \square , OCT \square) утверждается, что они выводят (высвечивают) на экран дисплея свое значение.

По директиве CLEAR символьные переменные не заменяются «изображениями (?) нулей». Им присваиваются пустые значения ($A \square = ""$, $LEN(A \square) = 0$).

Оператор COMMON не пересылает переменные в сопрягаемую (?) программу, а распределяет их в общей (транзитной) области памяти, содержимое которой доступно программе, сменяющей текущую.

Переменная CSRLIN не вызывает на экран дисплея значение вертикальной координаты курсора. Значением этой служебной переменной является номер строки, в которой сейчас находится курсор.

Оператор LPRINT USING предназначен для вывода на АЦПУ не только строчных констант. Основная его функция — форматный вывод значений арифметических и текстовых выражений.

Оператор REM не выделяет часть программы, которая не выполняется. Он сам принадлежит к невыполняемым операторам и используется для включения комментариев в текст исходной программы.

Сопоставление операторов цикла

по предусловию (WHILE—WEND) и операторов взаимодействия с подпрограммами (GOSUB—RETURN) вряд ли уместно. Оператор WHILE, выполняющий проверку предусловия, скорее можно уподобить оператору условного перехода IF. А оператор WEND, завершающий тело цикла, срабатывает как оператор безусловного перехода GOTO. В отличие от WEND оператор RETURN может передавать управление в разные точки.

Наконец, последняя группа замечаний касается синтаксических ошибок, допущенных в приводимых примерах. К ним относятся употребление запятой вместо десятичной точки (CINT (37, 92) COS (3, 14) — с. 94 и 95), употребление точки в конце тела оператора (операторы 20 на с. 103). Ряд ошибок в книге могут быть объяснены, увы, только недостаточной квалификацией автора или переводчика. Так, на с. 95 приведен пример установки текущей даты (DATE $\square \neq 7/19/83$). Он содержит две ошибки — неправильная запись знака присваивания и отсутствие кавычек при задании символьной константы (должно быть DATE $\square = „7/19/83“$). Затем утверждается, что на печати установленная дата будет выглядеть следующим образом: 08.29.1983. Откуда такая странная метаморфоза с номерами месяца и дня? Пример ввода символа с помощью служебной переменной INKEY \square должен быть таким: 30 X $\square = INKEY \square \cdot IF X \square = "" THEN 30 ELSE 40$. Только в этом случае он будет выполнять предписанную функцию.

Запятая между переменными A \square и B \square в операторе PRINT (с. 103) приведет к тому, что тексты TWO и FOR окажутся на экране разделенными довольно большим числом пробелов (запятая вызывает перемещение курсора в начало следующей зоны, которое отстоит от предшествующей границы на 15 позиций).

Канд. физ.-мат. наук Ю. Л. Кетков
(г. Горький)

Письмо Ю. Л. Кеткова, известного специалиста по языку Бейсик содержит серьезные замечания по качеству перевода седьмой главы.

Объем седьмой главы — 45 с.; все замечания, высказанные в письме, касаются первого раздела главы «Основные операторы» (с. 92—103), и практически все они справедливы. (Более того, список замечаний можно расширить.) Ниже разбираются изъяны книги, указанные в письме, в порядке их упоминания Ю. Л. Кетковым и анализируются возможные причины их появления.

1. Об операторе AUTO. Текст перевода: «При интенсивном программировании, чтобы получить правильный номер следующей строки, часто бывает необходимо постоянно ссылаться на номер предшествующей строки» — стилистически и содержательно слаб, слишком буквален. Причина этого в недостаточной осведомленности переводчика об особенностях процесса программирования на Бейсике. Речь в оригинале идет о том, что при вводе длинного текста программы (intensive pro-

gramming) приходится следить (refer) за тем, чтобы номер очередной строки получался увеличением номера предыдущей (в этом случае он будет «правильным»). Для автоматического увеличения номера строки и служит оператор AUTO.

2. Об операторе CHAIN. Первая из двух фраз об этом операторе просто повторяет инструкцию по Бейсику. Вторая, принадлежащая самому автору и вызвавшая возражения Ю. Л. Кеткова, — «это позволяет оператору переменные, введенные в одну программу, легко вставлять в другую...» — просто лишняя, и скорее путает, чем уточняет.

3. Об операторе LINE INPUT. Описание в книге следует инструкции по Бейсику (хотя и без должной строгости), но перевод сделан без какого-либо представления о реальной работе этого оператора и вследствие этого очень неточен. Несколько фактических ошибок переводчика, не указанных Ю. Л. Кетковым: «ограничения» вместо правильного «ограничители» (delimiters); «символьная

переменная, представляющая элемент массива, к которому строка должна быть присоединена» — должно быть: «символьная переменная или символьная переменная с индексом (array element), которой присваивается данная строка». Здесь же — опечатка: «256 знаков» вместо «255 знаков».

4. Об операторе LET. Ошибки переводчика: не «приписывает определенное значение выражению или переменной», а «присваивает переменной значение выражения». Вместо «дополнительным» должно быть «необязательным» (optional), причем необязательно слово LET, а не сам оператор. Поэтому последующая фраза, критикуемая автором письма, должна звучать примерно так: «Слово LET является необязательным, и его можно опускать для ускорения программирования».

5. Об операторе GOTO. Автор неправильно сопоставляет операторы GOTO и GOSUB: GOSUB так же безусловен, как GOTO; различие же их в том, что GOSUB осуществляет переход в подпрограмму с возвратом, а GOTO «безвозвратен».

6. Утверждения о том, что ряд функций (LEFT \square , RIGHT \square , MID \square , OCT \square , LOG, CSRLIN) выводят на экран свое значение, лежат на совести автора книги. Это явная небрежность изложения; из примеров видно, что вывод на экран происходит только в сочетании с оператором PRINT.

7. Об операторе CLEAR. Ошибка переводчика: вместо «изображения нулей» нужно «пустые значения» (null values), как это справедливо указывает Ю. Л. Кетков.

8. Об операторе COMMON. Снова Ю. Л. Кетков прав, а перевод буквальный и неточен: pass variables to a chained program означает не «переслать переменные в сопрягаемую программу», а передать (сделать доступными) переменные сменяющей программе.

9. Об операторе CSRLIN. Единственное место, где упрек автора письма не слишком основателен. «Вертикальная координата курсора» (так в инструкции по Бейсику и тексте перевода) это и есть «номер строки, в которой сейчас находится курсор» (так у Ю. Л. Кеткова). Имеются в виду текстовые координаты экрана: 25 строк по 40 или 80 колонок в каждой, нумеруемые соответственно сверху вниз и слева направо.

10. Об операторе LPRINT USING. Неточность (или небрежность) автора книги, оставшаяся в переводе и правильно отмеченная автором письма.

11. Об операторе REM. Неудачная формулировка автора книги, повторенная переводчиком. Впрочем, общий смысл оператора REM остается ясным. Слова «часть программы» нужно понимать как «часть текста программы, следующая за оператором REM до конца текущей строки».

12. Об операторе WHILE. Сопоставление операторов WHILE — WEND с операторами GOSUB — RETURN, которое проводится автором, совершенно поверхностно и примитивно и действительно неуместно. Здесь необходимо было дать примечание переводчика или редактора.

13. Опечатки. К тем, которые отмечены Ю. Л. Кетковым (пример с переменной DATE \square прямо-таки курьезен), можно добавить и другие: на с. 97 (2-я строка снизу) должно быть KEY 1, „LIST“; на с. 98 (в 14-й строке сверху) сразу две опечатки — B \square = LEFT (A \square .5) вместо B \square = LEFT \square (A \square .5).

Итак, изъяны перевода можно разделить на: — неточности оригинала, которые сохранились в процессе перевода (замечания 2, 6, 10, 11). Их, по всей видимости, следует оставить на совести автора книги и нельзя вменить переводчику в вину;

— ошибочные утверждения автора книги, которые нельзя было оставить без внимания и следовало снабдить примечаниями (чего сделано не было) — замечания 5 и 12;

— ошибки перевода, возникшие в результате слабого знакомства переводчика с языком Бейсик и, в особенности, с его версией для ПК ИБМ (замечания 1, 3, 4, 7, 8).

Легко понять, почему промахи в тексте перевода так обильно сосредоточились именно в этом небольшом разделе книги. Сообщаемая информация об операторах Бейсика чисто техническая. Она наполовину совпадает с официальным описанием языка в руководстве, выпущенной фирмой ИБМ. Вполне правильно перевести ее можно лишь при близком знакомстве с языком и компьютером, например, практически поработав на Бейсике ПК ИБМ.

Таким образом, проблемы, возникшие при переводе, вполне объяснимы, однако едва ли это оправдывает невысокое качество перевода вышеупомянутого раздела книги. Весьма поверхностное знакомство переводчика с программированием и, в частности, программированием на Бейсике, бросается в глаза, и можно было компенсировать этот пробел путем консультации по данному компьютеру и по языку программирования у специалистов. В период, когда делался перевод, это было достаточно реально: и компьютеры, и специалисты уже существовали.

Обсуждая перевод, нельзя оставить без внимания общий характер и качество оригинала, с которого он сделан. Сам автор, как видно из текста (не только этой главы), является, по всей видимости, программистом-любителем невысокой квалификации, без какого-либо основательного образования в этой области. Неудивительно, что раздел гл. 7 «Основные операторы», о котором шла речь, никак не может служить достоверной информацией о языке Бейсик. Ту же информацию можно с гораздо большим успехом и степенью достоверности почерпнуть из документации по языку. Автор прибегает просто-напросто к переложению инструкции в сокращенном виде, внося по ходу дела отдельные неточности и делая мало полезные добавления. (Справедливости ради, нужно сказать, что основная часть гл. 7 «Программирование на дисковой версии Бейсика» содержит наиболее оригинальную по сравнению с остальными главами книги информацию и представляет наибольший интерес для читателей).

Ю. Л. Кетков в своем письме не касается содержания других глав книги, хотя и дает понять, что для него они представляют незначительный интерес. Думается, что такой будет оценка большинства читателей.

Книга носит полурекламный характер и имеет сильный северо-американский отпечаток. В еще большей степени, чем гл. 7, другие главы насыщены мало оригинальными и притом чисто техническими сведениями, которые можно почерпнуть из документации или из специальной литературы и которые вряд ли когда-либо потребуются пользователю ПК ИБМ. (Так, в главе с лаконичным и интригующим названием «Графика» мы узнаем, между прочим, что «штырек 2 4-штырькового разъема платы адаптера не действует»).

Некоторые куски книги являются прямой перепечаткой других изданий. Многие приводимые данные успели сильно устареть (гл. 10).

Из сказанного видно, что объект перевода в свое время был выбран издательством «Мир» не слишком удачно: качество оригинала наложило на качество перевода и в итоге — издание вызывает справедливые нарекания заинтересованного читателя.

Ст. науч. состр. ВЦ АН СССР,
канд. физ.-мат. наук Г. В. Сенин

МИКРОПРОЦЕССОРНАЯ ТЕМАТИКА В КНИГАХ ГЛАВНОЙ РЕДАКЦИИ ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ ИЗДАТЕЛЬСТВА «НАУКА»

Для обучения и повседневной работы специалистов редакция подготовила большой выбор книг по программированию и применению вычислительной техники, в том числе микропроцессорных средств.

В серии «Библиотечка программиста» в 1987 г. будет издана работа Г. К. Боровина, М. М. Комарова и В. С. Ярошевского «Ошибки-ловушки при программировании на Фортране» о возникновении, обнаружении и предотвращении наиболее типичных ошибок, появляющихся при программировании на Фортране. Авторы объясняют методики поиска ошибок на ЭВМ различного типа и подробно разбирают задачи с наиболее типичными ошибками-ловушками.

В «Справочнике по алгоритмам и программам на Бейсике для персональных ЭВМ» В. П. Дьяконова (1987) изложены основы программирования на Бейсике — наиболее распространенном языке программирования для персональных ЭВМ (ПЭВМ). Основное внимание уделено алгоритмическому и программному обеспечению расчетов на ПЭВМ. В справочник включены около 300 программ на Бейсике для реализации основных численных методов, вычисления большинства специальных функций и решения ряда задач науки и техники. Даны краткие сведения об отечественных и зарубежных ПЭВМ и версии Бейсика для «Искры-226», ДВК-2, в том числе для микроЭВМ «Электроника 60», «Электроника ДЗ-28».

Несколько книг редакции посвящено обучению программированию.

Для всех, кто хотел бы быстро ознакомиться с основами программирования, в 1987 г. выйдет работа А. П. Ершова «Три урока по программированию» — достаточно строгое, но живое и иллюстрированное примерами изложение основ программирования.

Для учащихся средних школ, ПТУ, техникумов и начинающих программистов С. А. Абрамов и Е. В. Зима в книге «Начала программирования на языке Паскаль» (1987) предлагают сокращенный вариант Паскаля. Понятия языка и приемы программирования на нем излагаются так, чтобы с первых же параграфов привлечь читателя к самостоятельному разбору большого числа примеров, решению задач. Всякая программа на сокращенном Паскале будет правильной и в смысле полного варианта Паскаля.

В учебном пособии А. Г. Кушнirenко и Г. В. Лебедева «Программирование для математиков» на учебно-производственном языке высокого уровня излагается курс практического программирования для студентов математических специальностей вузов. Курс рассчитан на решение большого количества задач по изучению и модификации тщательно подготовленных текстов программ. Авторы приводят правила и примеры перекодировки программ на Фортран.

Для читателей, желающих приобрести навыки практического решения на ЭВМ задач, связанных с обработкой сложных структур данных, в 1986 г. издана монография В. Н. Касьянова и В. К. Сабельфель-

да «Сборник заданий по практикуму на ЭВМ».

Тем, кто применяет ЭВМ в робототехнике, адресованы книги Г. С. Черноруцкого, А. П. Сибрина и В. С. Жабреева «Следящие системы автоматических манипуляторов», Е. П. Попова «Робототехника и гибкие производственные системы» (1987) и В. А. Бесекерского, В. В. Изранцева «Системы автоматического управления с микроЭВМ» (1987). Авторы последней рассматривают схемотехнику микроЭВМ и особенности программирования, большое внимание уделяют реализации на базе микроЭВМ цифровых устройств для контроля и повышения надежности систем управления. Вопросам проектирования самодиагностирующих модулей вычислительных устройств и практическим приемам создания устройств с малой трудоемкостью тестового диагностирования посвящена монография А. П. Горяшко «Синтез диагностируемых схем вычислительных устройств» (1987).

В серии «Проблемы искусственного интеллекта» вышла монография Д. А. Поспелова «Ситуационное управление: теория и практика» (1986) о методе управления сложными техническими и организационными системами, основанном на идеях теории искусственного интеллекта.

Под редакцией А. П. Ершова выйдет монография Д. Я. Левина «Инструментальный комплекс программирования на основе языков высокого уровня» (1987). Комплекс включает в себя теоретико-множественный язык программирования сверхвысокого уровня СЕТТЛ, языки управления базами данных, средства построения виртуальных машин, продукционные языки.

В этой же серии в 1987 г. будут изданы сборник статей «Моделирование языковой деятельности в интеллектуальных системах» и научно-популярная работа И. Н. Горелова «Разговор с компьютером: психолингвистические аспекты проблемы». Обе книги посвящены проблеме общения ЭВМ с человеком на естественном языке. В монографии Э. В. Попова «Экспертные системы: решение неформализованных задач в диалоге с ЭВМ» (1987) рассматриваются проблемы создания систем не только общающихся с человеком на естественном языке, но и моделирующей деятельность эксперта при решении неформализованных задач. В книге систематически излагаются принципы построения экспертных систем, их основные компоненты, состав и способы изложения знаний. Автор уделяет особое внимание способностям экспертных систем приобретать новые знания, объяснять свои действия, автоматически формировать решение задачи, общаться на естественном языке и приводит примеры широкого использования экспертных систем в различных предметных областях.

С. С. Матвеев



1р. 10к.

 **МП**
Индекс 70588

 **МП**



ВНИМАНИЕ

Напоминаем, что журнал издается шесть раз в год!

Подписка на него поквартальная и принимается всеми отделениями Союзпечати в течение всего года!

За месяц до начала любого квартала прекращается подписка на ту часть годового комплекта журналов, которая издается в следующем трехмесячном периоде.

Цена одного номера — 1 руб. 10 коп.

Индекс журнала — 70588.