

Микро- КНИГА 6 ЭВМ

Универсальные
машины
семейства СМ 1800



Универсальные машины семейства СМ 1800

1

Принципы построения
семейства микроЭВМ
СМ 1800

2

Программное
обеспечение
семейства микроЭВМ
СМ 1800

3

Построение
комплексов моделей
семейства СМ 1800
и их применение

4

МикроЭВМ
Искра-226

Приложение
Универсальный
программный
интерфейс

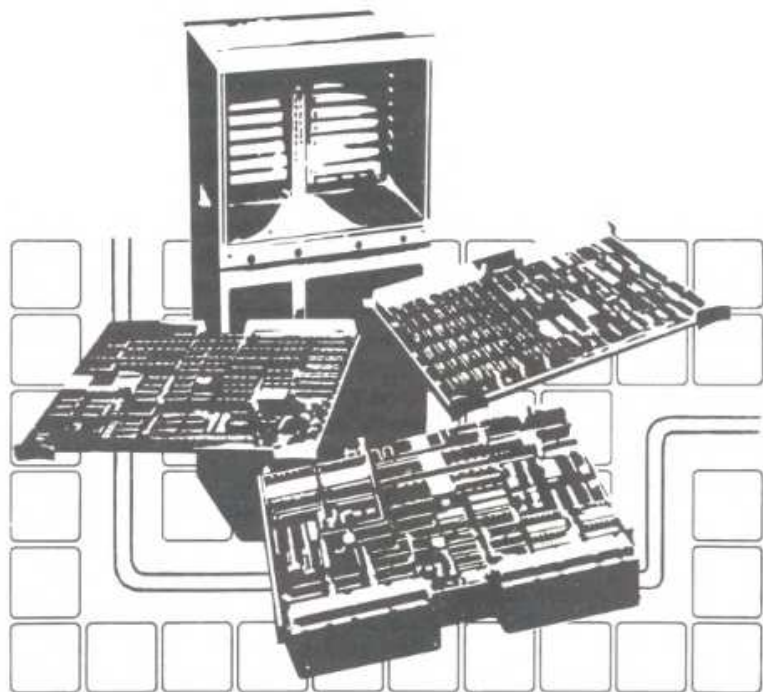
Заключение

Перспективы развития
универсальных
микроЭВМ

МикроЭВМ

В ВОСЬМИ КНИГАХ

Под редакцией
лауреата Государственной премии СССР
члена-корреспондента АН СССР
Л. Н. ПРЕСНУХИНА



**Универсальные
машины
семейства СМ 1800**

КНИГА

6



Москва «Высшая школа» 1988



Scan AAW

ББК 32.973.2

М59

УДК 681.322

Рекомендовано Министерством высшего и среднего специального образования СССР для использования в учебном процессе

Н. Д. Кабанов, В. С. Кравченко, А. Н. Шкамарда, Д. И. Панфилов

Рецензенты: кафедра «Автоматизированные системы управления» Московского высшего технического училища им. Н. Э. Баумана (зав. кафедрой — проф. В. Н. Четвериков); лауреат Государственной премии СССР канд. техн. наук Б. И. Ермолаев (Научно-исследовательский центр электронной вычислительной техники)

МикроЭВМ: В 8 кн.: Практик. пособие/Под ред. М59 Л. Н. Преснухина. Кн. 6. Универсальные машины семейства СМ 1800/Н. Д. Кабанов, В. С. Кравченко, А. Н. Шкамарда, Д. И. Панфилов. — М.: Высш. шк., 1988. — 158 с.: ил.

В пособии рассмотрены общие структурные свойства и особенности выполнения микроЭВМ и контроллеров, построенных на микроЭВМ серии СМ 1800, СМ 1810 и «Искра-226»; элементная база, архитектура, принципы организации и программное обеспечение семейства микроЭВМ, примеры построения и применения информационно-измерительных, вычислительных и управляющих комплексов, показаны перспективы развития и совершенствования микроЭВМ.

М $\frac{2405000000-394}{001(01)-88}$ 150-88

ББК 32.973.2
6Ф7.3

© Издательство «Высшая школа», 1988

Введение

В настоящей книге рассматривается один класс микропроцессорных средств — универсальные микроЭВМ. По назначению эти микроЭВМ отличаются от микроЭВМ других классов широким диапазоном применений, по способу их построения — гибкостью внутренней структуры, функциональной модульностью и возможностью наращивания технических и программных средств. В состав универсальных микроЭВМ, как правило, входят модуль центрального процессора, модули связи с периферией и различные устройства ввода — вывода, модули связи с другими ЭВМ или системами, а также связи с объектами управления. Системное программное обеспечение строится по модульному принципу. Пример универсальной микроЭВМ ведущей западной фирмы — модульный набор фирмы «Intel», США [1].

Рассмотрим некоторые тенденции развития микроЭВМ этого класса. С развитием элементной базы и увеличением степени интеграции БИС происходит наращивание вычислительной мощности как отдельных компонентов микроЭВМ — функциональных модулей, так и микроЭВМ в целом. Сами модули представляют собой сложные многоуровневые вычислительные структуры. В свою очередь, для надежной связи между отдельными модулями или группами модулей необходима гибкая, высокоскоростная (десятки М байт/с) система связи, что приводит к необходимости организации более мощной структуры внутренней функции взаимодействия — внутреннего или системного интерфейса микроЭВМ. При этом получают дальнейшее развитие средства ввода — вывода. В результате освоения новых технологий появляются новые способы и средства ввода — вывода информации. Например, плазменные или люминесцентные

устройства вывода с цветным отображением алфавитно-цифровой или графической информации, устройство ввода — вывода человеческого голоса, высокоскоростные принтеры и т. д. Изменяются средства межмашинных (или системных) связей. Появляются локальные оптоволоконные сети, средства связи с использованием инфракрасных лучей. Организуются глобальные сети спутниковой связи, где в качестве терминальных устройств используются также универсальные микроЭВМ. Таким образом, совершенствуются средства и одновременно расширяются области их применения.

Отечественной промышленностью в соответствии с программой системы мини-ЭВМ (СМ ЭВМ) в 1981 г. освоен выпуск 8-разрядных универсальных микроЭВМ семейства СМ 1800 [2]. В настоящее время в его состав входит более 15 моделей, имеющих разнообразные применения — от простых контроллеров для встраивания в оборудование до сложных многоуровневых мультипроцессорных систем. Проводятся разработки еще нескольких моделей, представляющих собой 16-разрядные микроЭВМ, выполненные в различных конструктивных исполнениях и предназначенные в основном для использования в управляющих системах.

В настоящей книге описываются назначение, архитектура, технические и программные средства, область применения универсальных микроЭВМ семейства СМ 1800.

Авторы выражают благодарность своим коллегам по разработкам семейства микроЭВМ СМ 1800 — В. Д. Гуськову, В. А. Соболеву, И. М. Рыбину, М. С. Белоконову, В. Г. Каневскому, И. И. Бабанову, А. Д. Азарову, И. С. Моросанову и Л. А. Казарян.

Принципы построения семейства микроЭВМ СМ 1800



Семейство микроЭВМ СМ 1800 включает в себя следующие типы микроЭВМ: СМ 1800, СМ 1804, СМ 1810 и СМ 1814.

МикроЭВМ СМ 1800 и СМ 1804 построены на базе 8-разрядного микропроцессора КР580ИК80А. МикроЭВМ СМ 1810 и СМ 1814 построены на базе 16-разрядного микропроцессора К1810ВМ86.

МикроЭВМ СМ 1800 и СМ 1810 имеют обычное исполнение для нормальных условий эксплуатации. МикроЭВМ СМ 1804 и СМ 1814 выполнены в пылезащищенном исполнении и могут работать в тяжелых промышленных условиях.

В основу построения семейства микроЭВМ СМ 1800 положен модульный принцип, суть которого состоит в том, что машины выполняются из функционально законченных устройств (модулей) в виде одной или двух печатных плат, объединенных интерфейсом И41. Универсальность интерфейса И41 и модульность ЭВМ позволяют комплексовать микропроцессорные системы на уровне печатных плат, что обеспечивает высокую гибкость при создании управляющих и вычислительных комплексов, а также открытость микроЭВМ как системы. Открытость микроЭВМ создает условия для расширения номенклатуры модулей и развития СМ 1800. Так, при создании микроЭВМ СМ 1810 использованы новые элементная база и конструктивная реализация. Однако открытость семейства позволила использовать в микроЭВМ СМ 1810 практически все модули из состава СМ 1800. Большое число серийно выпускаемых модулей, каждый из которых имеет свой шифр и цену, позволяет создавать экономичные и надежные микропроцессорные системы для конкретных применений. Получение требуемой производительности обеспечивается использованием модулей центрального процессора различной производительности и имеющейся возможностью построения мультимикропроцессорных систем. Архитектура микроЭВМ СМ 1800 допускает одновременную работу до 8 модулей центрального процессора, а СМ 1810 — до 16. Каждый модуль центрального процессора содержит микропроцессор, локальную оперативную и постоянную память, связь с интерфейсом И41. Системный интерфейс И41 подробно описан в [2].

1.1. Элементная база

Основу элементной базы семейства микроЭВМ СМ 1800 составляют микропроцессорные наборы КР580 и К1810.

Микропроцессорный набор КР580 включает в себя интегральные схемы согласно табл. 1.1.

Таблица 1.1

Тип микросхемы	Зарубежный аналог	Основная функция
КР580ИК80А	18080А	Микропроцессор 8-разрядный
КР580ГФ24	18224	Генератор тактовых импульсов
КР580ИК55	18255	Программируемый параллельный интерфейс для периферийных устройств
КР580ИК51А	18251А	Программируемый последовательный интерфейс связи
КР580ВИ53	18253	Программируемый таймер интервалов
КР580ВГ18	18218	Контроллер магистрали
КР580ВИ59	18259	Контроллер прерываний
КР580ВК28	18228	Системный контроллер
КР580ВК38	18238	Системный контроллер
КР580ИР82	18282	Буферные регистры 8-разрядные
КР580ИР83	18283	
КР580ВА86	18286	Формирователи магистрали 8-разрядные
КР580ВА87	18287	

Микропроцессорный набор К1810 включает в себя интегральные схемы согласно табл. 1.2.

Таблица 1.2

Тип микросхемы	Зарубежный аналог	Основная функция
К1810ВМ86	18086	Микропроцессор 16-разрядный
КМ1810ВМ89	18089	Микропроцессор ввода — вывода
КР1810ГФ84	18284	Тактовый генератор
КР1810ВГ88	18288	Контроллер магистрали
КР1810ВИ59А	18259А	Контроллер прерываний
КР1810ВБ89	18289	Арбитр магистрали

Микропроцессорные наборы КР580 и К1810 подробно описаны в кн. 3 настоящей серии.

Микропроцессорный набор К1810 представляет собой третье поколение микропроцессоров. Он расширяется арифметическим сопроцессором, аналогичным микропро-

цессору 18087 фирмы «Intel» (США), и процессором ввода — вывода КМ1810ВМ89. Модульная структура набора К1810 базируется на трех основных принципах:

- 1) основные функции проектируемой системы распределены между специальными элементами;
- 2) многопроцессорный режим работы проектируемой системы реализуется аппаратно;
- 3) иерархическая структура интерфейса обеспечивает эффективную обработку потока данных в высокопроизводительных системах.

Основной микропроцессор К1810ВМ86 имеет следующие характеристики:

Тактовая частота	— 5 МГц
Корпус	— 40-контактный
Разрядность данных	— 16
Диапазон адресации	— до 1 М байт для памяти, до 64 К байт для устройства ввода — вывода
Передача данных и адреса	— по одной физической линии с мультиплексированием по времени, т. е. передача дан- ных производится после пе- редачи адреса

Микропроцессор К1810ВМ86 имеет два режима работы: минимальный и максимальный.

В минимальном режиме микропроцессор формирует управляющие сигналы для памяти и устройств ввода — вывода, обеспечивая однопроцессорный режим работы.

В максимальном режиме сигналы управления памятью и устройствами ввода — вывода формируются контроллером КР1810ВГ88 на основании информации о состоянии микропроцессора. Линии, используемые в минимальном режиме для управления памятью и устройствами ввода — вывода, в максимальном режиме обеспечивают аппаратную реализацию многопроцессорного режима работы.

Сопроцессор 18087 представляет собой 40-контактную БИС и предназначен для выполнения арифметических операций с плавающей запятой. При отсутствии его операции с плавающей запятой выполняются программным способом, что существенно снижает производительность системы.

Микропроцессор ввода — вывода КМ1810ВМ89 обычно работает под управлением центрального процессора

K1810BM86, но может использоваться самостоятельно в контроллерах ввода — вывода. Он имеет два независимых программно-управляемых канала ввода — вывода, которые осуществляют обмен данными по прямому доступу в память со скоростью до 1,25 М байт/с. Сочетая высокую скорость передачи информации с программируемой логикой, микропроцессор КМ1810BM89 повышает эффективность системы, освобождая центральный процессор от рутинных операций по обработке информации устройств ввода — вывода.

Контроллер прерываний К1810ВИ59А предназначен для обработки сигналов прерываний, поступающих от периферийных устройств, и работает совместно с микропроцессором К1810BM86. Он принимает запросы на прерывание от восьми источников. При каскадном подключении дополнительных контроллеров К1810ВИ59А число источников можно увеличить до 64. Уровень приоритета прерываний задается программой от центрального процессора. Контроллер выделяет запрос на прерывание с высшим приоритетом и, после того как центральный процессор примет запрос на прерывание, передает код, определяющий конкретный источник прерывания.

Тактовый генератор К1810ГФ84 предназначен для формирования сигналов частотой 5 МГц, синхронизирующих работу микропроцессора К1810BM86. Кроме того, КР1810ГФ84 формирует сигналы СБРОС и ГОТОВНОСТЬ.

Буферные 8-разрядные регистры КР580ИР82 и КР580ИР83 используются для демультимплексирования магистрали адреса — данных микропроцессора К1810BM86, что необходимо для большинства периферийных контроллеров и интерфейса И41.

Формирователи магистрали 8-разрядные КР580ВА86 и КР580ВА87 предназначены для обеспечения необходимой мощности интерфейсных сигналов, выходящих за пределы платы. Они представляют собой биполярные приемопередатчики с трехстабильными выходами. Формирователь КР580ВА87 в отличие от КР580ВА86 инвертирует входные сигналы.

Контроллер магистрали КР1810ВГ88, предназначенный для декодирования байта состояния микропроцессора КМ1810BM89 или К1810BM86 (в максимальном режиме), осуществляет генерацию во времени команд и управляющих сигналов для системной магистрали. Он также выдает сигнал стробирования адреса в буфер-

ные регистры КР580ИР82 и КР580ИР83 во время демультиплексирования адреса — данных от микропроцессора К1810ВМ86.

Арбитр шины КР1810ВБ89 предоставляет системную магистраль одному из нескольких задатчиков, которые выставили запрос на захват магистрали для доступа к ресурсам системы, например к общей памяти.

Многопроцессорная система, построение которой обеспечивает микропроцессорный набор К1810 и интерфейс И41, имеет следующие преимущества перед однопроцессорной:

- системная задача делится на подзадачи, каждая из которых решается специально для нее спроектированным процессором;

- несколько процессоров могут работать одновременно, повышая общую производительность системы;

- увеличивается надежность путем распределения функций системы между процессорами так, что отказ одной части не влечет за собой выход из строя или остановку всей системы;

- модульная структура увеличивает гибкость, т. е. модификация системы возможна добавлением или заменой отдельных модулей, а не переработкой всей системы.

Архитектура семейства К1810 решает две проблемы многопроцессорной обработки: арбитраж и монопольное владение магистралью. Арбитр КР1810ВБ89 предоставляет магистраль процессору после ее освобождения другими задатчиками. Монопольное владение предполагает, что процессору необходимо осуществить захват магистрали на продолжительное время — монопольный режим работы, при этом процессор выдает сигнал блокировки, который активизируется программно, для запрета доступа к магистрали с других процессоров. В этом режиме скорость обмена информацией увеличивается за счет исключения времени, требуемого для арбитража. Монопольный захват магистрали обычно используется при работе с жесткими магнитными дисками, когда требуется большая скорость передачи информации, и при использовании памяти в режиме «почтового ящика». При записи или чтении массива информации одним процессором недопустимо вмешательство других процессоров или устройств ввода — вывода, так как не полностью записанный массив является ложным для пользователей. Монопольный захват шины требует внимания от системного программиста, который проектирует программу, ра-

ботающую в реальном масштабе времени, например программу управления технологическим процессом. Он должен учитывать, что на время монопольного захвата ни одно устройство ввода — вывода, подсоединенное к этой магистрали, не может сообщить процессору о своем состоянии, т. е. время реакции системы на аварийную ситуацию не может быть меньше времени монопольного захвата магистрали.

На рис. 1.1 показана типовая модульная структура системы, которую можно построить на базе микропроцессорных наборов К580 и К1810. Как видно из рисунка, возможны системные и локальные магистрали, объединяющие несколько процессоров. Микропроцессоры всегда связаны с локальной магистралью, а запоминающие устройства, устройства ввода — вывода и модули процессора — с системной магистралью.

Локальная магистраль реализуется микропроцессором К1810ВМ86 и координирует работу нескольких процессоров и сопроцессоров. Так, в модуле центрального процессора МЦП-16 микроЭВМ СМ 1810 локальная магистраль связывает микропроцессор К1810ВМ86 и сопроцессор 18087.

Системная магистраль состоит из пяти наборов сигналов: адреса, данных, управления, прерывания, арбитража.

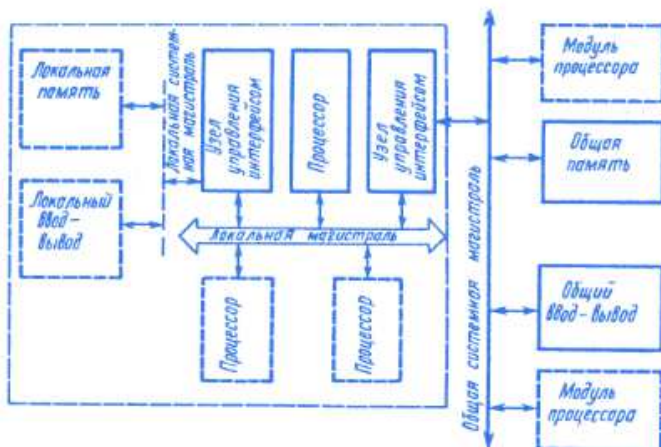


Рис. 1.1. Типовая структура системы на базе микропроцессорных наборов К580 и К1810

Интерфейс И41 — пример общей системной магистрали, которая позволяет координировать работу модулей, выполняющих различные функции [2].

Возможна также локальная системная магистраль, не выходящая за пределы модуля процессора, но позволяющая подсоединять дополнительные устройства непосредственно к микропроцессору. Она недоступна со стороны модулей системы, объединенных общей системной магистралью. Локальная системная магистраль разгружает общую и позволяет процессорному модулю осуществлять обмен информацией со своими устройствами, освобождая общую системную магистраль для работы с другими модулями.

В микроЭВМ СМ 1810 роль локальной системной магистрали выполняет локальная шина модуля МЦП-16, через которую осуществляется связь с локальной памятью объемом 256 К байт и периферийными устройствами.

Таким образом, процессоры, узлы управления интерфейсами, объединенные локальной магистралью, и устройства, объединенные локальной системной магистралью (в данном случае локальная память и локальный ввод — вывод), образуют модуль процессора. Простейший модуль состоит из микропроцессора, узла управления локальным интерфейсом, объединяющим локальную память и управление локальными устройствами ввода — вывода. Описанная структура построения используется при проектировании одноплатных ЭВМ, не имеющих возможности для расширения.

На рис. 1.1 показан модуль процессора, предназначенный для работы в расширяемой модульной многопроцессорной системе. Узлы управления общей системной магистралью обеспечивают доступ модуля к общей памяти и обмен информацией с другими процессорными модулями. Если в системе имеется несколько процессоров, то все запоминающие устройства и устройства ввода — вывода, подсоединенные к общей системной магистрали, доступны процессорным модулям. Арбитры магистрали КР1810ВВ89 в каждом модуле процессора обеспечивают доступ модулей к общей системной магистрали и, следовательно, к общей памяти и устройствам ввода — вывода.

Узел управления локальной системной магистралью обеспечивает индивидуальный доступ процессора к собственной памяти и устройствам ввода — вывода, недос-

тупные для других процессорных модулей. При этом проектировщик системы может распределять местные и общие ресурсы таким образом, чтобы обеспечить необходимую надежность и производительность системы. Например, для обеспечения надежности особо важную информацию можно хранить как в общей, так и в локальной памяти. Для обеспечения производительности можно включить в систему несколько процессорных модулей, каждый из которых будет решать частную задачу в заданное время. При этом обеспечивается быстрое распределение общедоступных ресурсов системы.

1.2. Архитектура микроЭВМ

Микропроцессорные наборы К580 и К1810 позволяют проектировать системы различной сложности, начиная с минимальных, состоящих из 4—5 БИС, и кончая многопроцессорными с распределением функций между процессорными модулями и работающими с локальной и общей памятью, с локальными и общими устройствами ввода — вывода.

Схема реализации минимального режима работы показана на рис. 1.2.

Периферийные компоненты, подсоединяемые к микропроцессору, должны иметь встроенные средства демultipлексирования адреса — данных. Часто из экономических соображений эти компоненты выполняют многофункциональными, сочетая, например, оперативное и постоянное запоминающие устройства, порты ввода — вывода и таймер на одном кристалле. Используя эти компоненты, можно создать небольшие (например, из 4 БИС) экономичные системы для решения определенных задач.

Значительно чаще используется минимальный режим работы с разделением магистралей адреса — данных. На рис. 1.3 показана схема реализации этого режима. Два буферных регистра КР580ИР82/83 позволяют получить доступ к памяти объемом 64 К байт. Добавив третий буферный регистр, можно получить доступ к памяти объемом 1 М байт. Этот режим работы позволяет эффективно использовать компоненты из микропроцессорного набора К580. Формирователи КР580ВА86/87 могут объединять по схеме ИЛИ магистрали данных от нескольких источников информации. Контроллер прерываний КР1810ВИ59А обеспечивает реакцию процессора на



Рис. 1.2. Схема реализации минимального режима работы



Рис. 1.3. Схема реализации минимального режима работы с разделением магистралей адреса — данных

любой из восьми возможных запросов на прерывание, при этом не требуется опрашивать регистры состояний источников прерываний, так как контроллер формирует код переданного микропроцессору источника прерывания. Минимальный режим работы набора К1810 широко применяется при проектировании персональных ЭВМ.

Максимальный режим работы микропроцессора К1810ВМ86 используется при построении многопроцессорных систем. На рис. 1.4 показана схема реализации максимального режима работы микропроцессора К1810ВМ86, когда многопроцессорность обеспечивается с использованием микропроцессоров ввода — вывода КМ1810ВМ89.

Контроллер шин КР1810ВГ88 формирует сигналы управления из байта состояния микропроцессора К1810ВМ86, освобождая линии для организации работы нескольких микропроцессоров. Встроенные в микропроцессоры

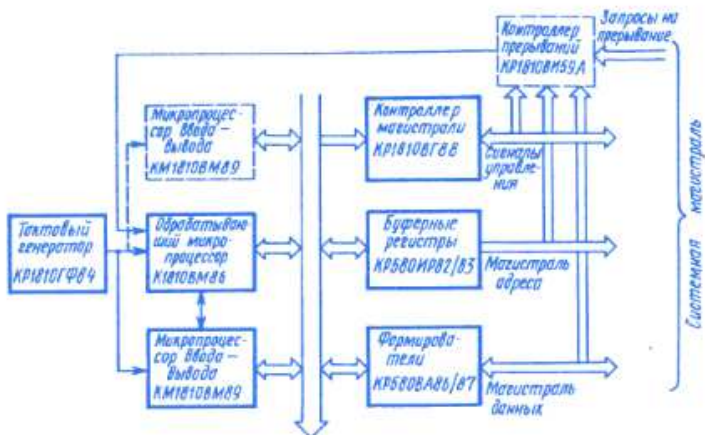


Рис. 1.4. Схема реализации максимального режима с возможностью работы нескольких микропроцессоров

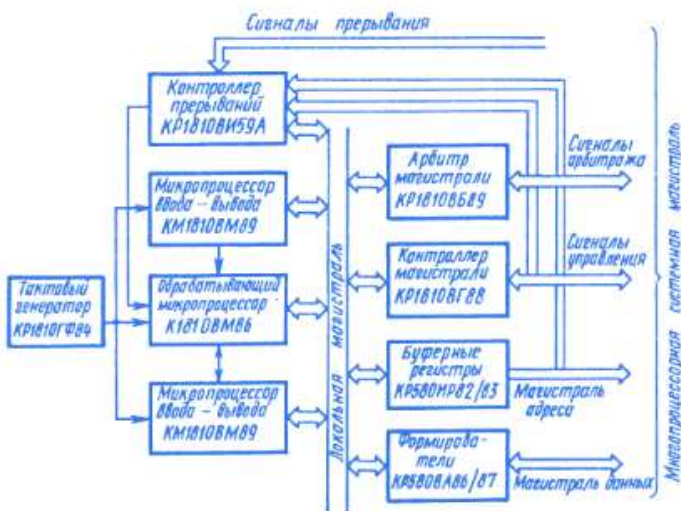


Рис. 1.5. Схема реализации максимального режима с возможностью работы нескольких модулей процессора

К1810ВМ86 и КМ1810ВМ89 схемы арбитража координируют пользование локальной и системной магистралями.

Подобная структура используется в системах с одним процессорным модулем для эффективной работы устройств ввода — вывода. Например, один микропроцессор КМ1810ВМ89 используется для управления накопителем на магнитных дисках, а другой — для управления накопителем на магнитной ленте.

На рис. 1.5 показана схема реализации максимального режима работы микропроцессора К1810ВМ86, когда многопроцессорность обеспечивается на уровне как микропроцессоров, так и процессорных модулей. Подключение арбитра магистрали КР1810ВБ89 позволяет спроектировать многопроцессорную системную магистраль, координирующую работу нескольких процессорных модулей. Каждый процессорный модуль имеет локальную магистраль, к которой подсоединяются один микропроцессор К1810ВМ86 и один или два микропроцессора КМ1810ВМ89. Все процессоры, объединенные локальной магистралью, имеют доступ к системной магистрали через одни и те же микросхемы.

На рис. 1.6 показан пример структуры, позволяющий организовать работу локальной магистрали ввода —

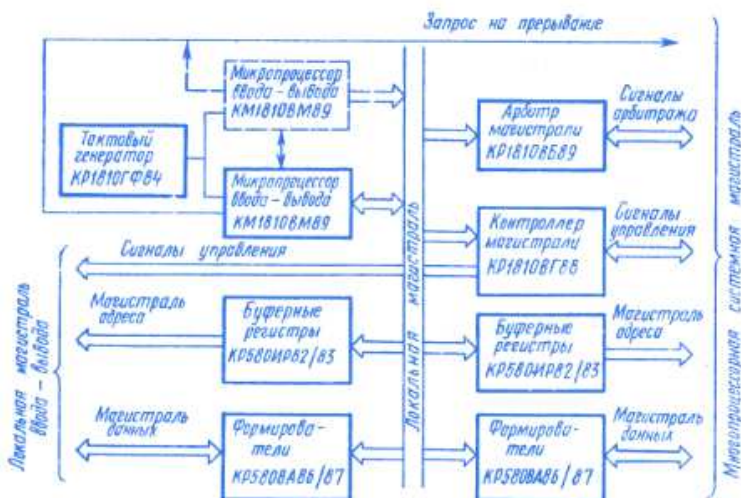


Рис. 1.6. Структурная схема реализации работы локальной магистрали ввода — вывода и многопроцессорной системной магистрали

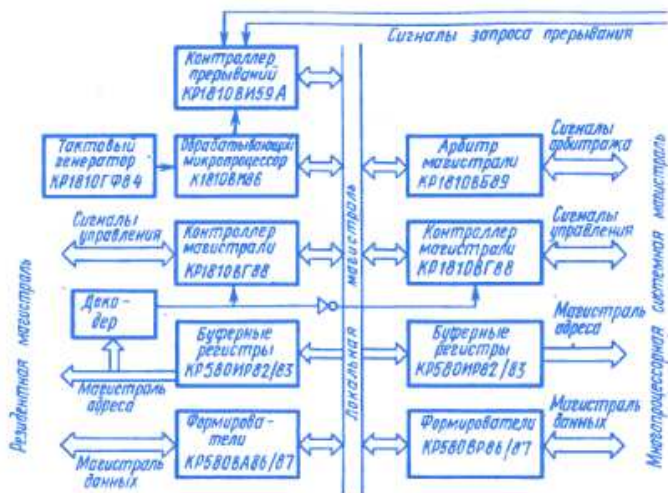


Рис. 1.7. Структура процессорного модуля с резидентной магистралью

вывода и многопроцессорной системной магистральной. В данной структуре доступ к памяти осуществляется через системную магистраль, а доступ к устройствам ввода — вывода — через локальную магистраль. Для повышения производительности системы память, в которой хранятся программы микропроцессора КМ1810ВМ89, подсоединяется к локальной магистральной ввода — вывода, что позволяет разгрузить системную магистраль. Такая структура часто используется при проектировании контроллеров ввода — вывода, например контроллеров накопителей на магнитных дисках.

На рис. 1.7 показана структура процессорного модуля с резидентной магистралью, к которой можно подсоединить локальную память с собственным адресным пространством. Декодер, расшифровывая адрес, распознает, в системную или резидентную магистраль следует выдавать сигналы управления. При добавлении второго арбитра КР1810ВБ89 можно получить резидентную магистраль с параметрами системной, т. е. мультипроцессорную резидентную магистраль.

Таким образом, микропроцессорные наборы К580 и К1810 позволяют проектировать мультимикропроцессорные системы различной сложности. При этом каждый

модуль проектируется оптимально для решения конкретной задачи. Модули могут взаимодействовать друг с другом посредством как прерываний, так и обмена сообщениями, размещенными в памяти коллективного пользования. Система может расширяться добавлением модулей с новыми функциями.

На рис. 1.8 показана гипотетическая система с девятью микропроцессорами, состоящая из пяти процессорных модулей и одного модуля системной памяти. На рисунке компоненты связи с магистралями не показаны.

Модуль-диспетчер распределяет задания среди других процессорных модулей и выполняет программу, которая находится в локальной памяти и недоступна другим модулям. Системная память доступна всем модулям и используется для обмена сообщениями, организации общих буферов и хранения информации коллективного пользования. Все это повышает защищенность процессорных модулей друг от друга и минимизирует конфликтные ситуации.

Модуль базы данных отвечает за организацию системных файлов, поступающих с устройств ввода — вывода.

Каждый из трех модулей управления графическим терминалом обеспечивает реализацию того графического стандарта, который принят для данного терминала. Микропроцессор ввода — вывода КМ1810ВМ89 в каждом модуле обеспечивает связь устройств ввода — вывода, включая электронно-лучевую трубку, с микропроцессором К1810ВМ86 (в модуле базы данных — с микропроцессором КМ1810ВМ89), выполняющим функции обработки информации по программам, хранящимся в локальной памяти.

Архитектура, принятая в семействе микроЭВМ СМ 1800, оптимально использует все возможности, предоставляемые микропроцессорными наборами К580 и К1810, для эффективной организации вычислительного процесса в проектируемых системах. Прежде всего это определяется многопроцессорным интерфейсом И41 [2], позволяющим скоординировать работу 16 процессорных модулей. В микроЭВМ СМ 1810 интерфейс И41 расширен (по отношению к описанному в [2]) следующим образом:

- разрядность шины адреса увеличена с 20 до 24, что обеспечивает адресацию памяти до 16 М байт;
- добавлен сигнал блокировки LOCK для организации монопольного захвата магистрали.

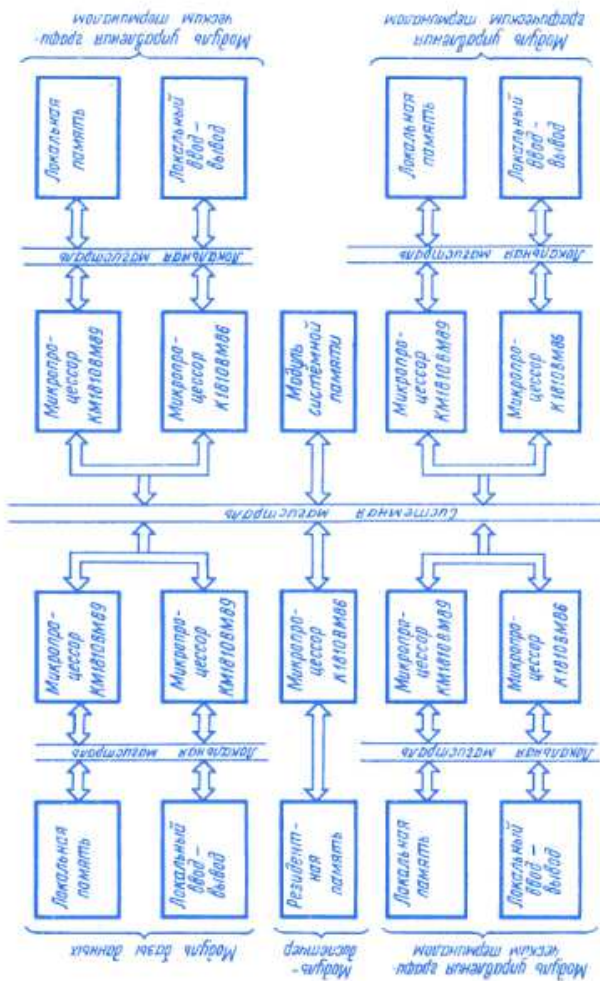


Рис. 1.8. Структура управления тремя графическими дисплеями

Основная особенность интерфейса И41 состоит в том, что микропроцессорные наборы К580 и К1810 оптимальны для его физической реализации. Описанные свойства многопроцессорной системной магистрали присущи и интерфейсу И41. Кроме того, интерфейс И41 имеет в своем составе сигналы, позволяющие автоматически подавать резервное питание оперативной памяти в случае исчезновения основного питания. Это свойство позволяет сохранять информацию в полупроводниковом запоминающем устройстве при пропадании сетевого напряжения, что возможно в реальной системе управления технологическим процессом. Общая структура микроЭВМ СМ 1810 приведена на рис. 1.9, где все модули соединены через единый интерфейс И41 в соответствии с протоколом обмена (описанным в [2]). В любой операции обмена участвуют два модуля: задатчик и исполнитель. Исполнителем может быть любой модуль, подключенный к интерфейсу. Задатчиком может быть только модуль, имеющий средства для захвата интерфейса, т. е. арбитр магистрали типа КР1810ВВ89. Одновременно два задатчика и более, естественно, не могут захватить магистраль. Конструкция микроЭВМ СМ 1810 и интерфейс И41 позволяют любой модуль поместить в любое место машины, что упрощает сборку системы для конкретного пользователя. Только модуль системного контроля МСК-16 устанавливается в определенное место машины и обязательно присутствует, так как он выполняет функции, связанные со всей системой. В частности, функции арбитража запросов на владение магистралью выполняются МСК-16.

Если одновременно несколько задатчиков запрашивают разрешение на управление магистралью, то вопрос о том, какому задатчику будет предоставлена магистраль, решается арбитром в соответствии с приоритетами задатчиков.

В машине реализуется параллельная или циклическая схема арбитража.

При параллельном приоритетном арбитраже степень приоритетности для задатчиков задается их месторасположением в монтажном блоке.

При циклическом арбитраже приоритеты задатчиков циклически изменяются — задатчик, завершивший обращение к интерфейсу, приобретает низший приоритет, следующий задатчик — высший приоритет. Необходимый вариант арбитража задается переключкой на наборном

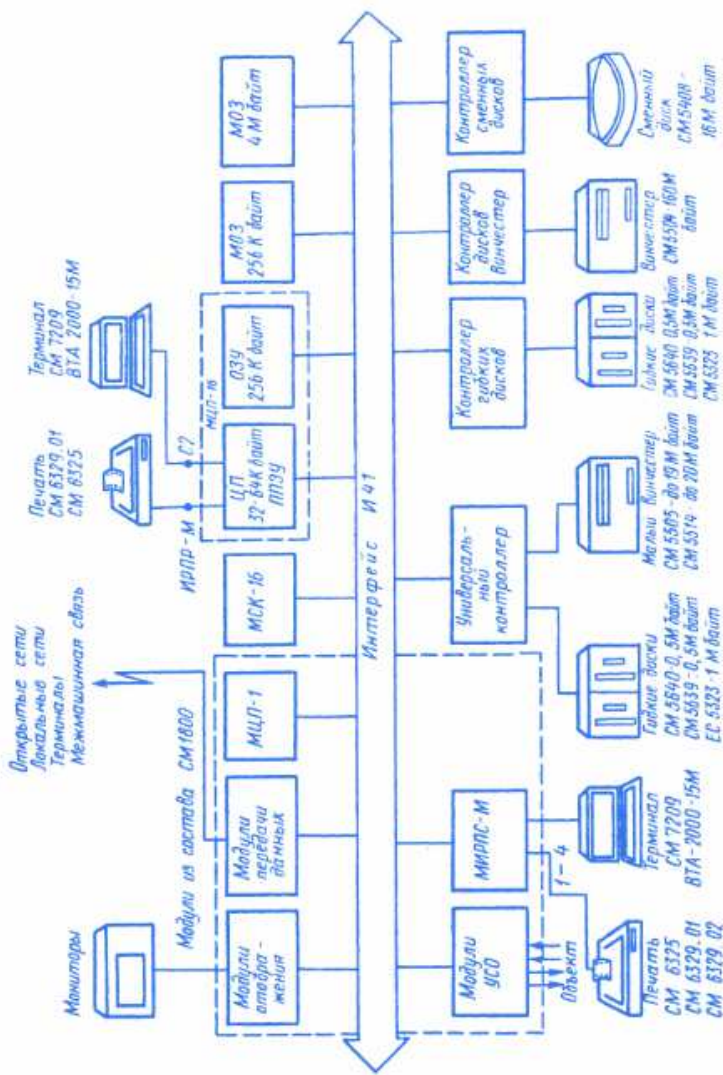


Рис. 1.9. Общая структура микроЭВМ SM 1810

поле модуля МСК-16 (отсутствие переключки соответствует параллельному арбитражу, наличие — циклическому). Имеется 16 уровней приоритета, к каждому уровню подключается только одно устройство (модуль). Модуль центрального процессора подключается к уровню с наименьшим приоритетом. Остальные 15 уровней используются другими процессорами, а также устройствами (модулями), которые занимают магистраль для обмена информацией с памятью, минуя процессор (прямой доступ к памяти).

Каждый уровень приоритета имеет линии запроса и разрешения прямого доступа. Запрос прямого доступа любого уровня приостанавливает работу центрального процессора даже во время выполнения им команды программы.

Управляя сигналом LOCK, модуль центрального процессора (МЦП) блокирует захват магистрали, т. е. работает в монопольном режиме.

Кроме работы по запросам, т. е. в режиме прямого доступа, возможен обмен данными между МЦП и другими модулями по сигналам прерывания. Для обмена данными модуль выдает сигнал прерывания, по которому МЦП прерывает основную программу и переходит на подпрограмму обслуживания этого модуля. Всем запросам на прерывание присвоен определенный приоритет. Интерфейс И41 имеет восемь уровней приоритета прерываний, каждому из которых соответствует своя линия запроса на прерывание. К каждой линии запроса на прерывание подключается несколько модулей.

Запросы на прерывание поступают в МЦП в произвольные моменты времени, но воспринимаются им по завершении текущей команды программы. При этом МЦП определяет наиболее приоритетный из поступивших запросов и переходит на подпрограмму обработки прерывания. При этом в стеке сохраняется адрес следующей команды прерванной программы, в конце подпрограммы этот адрес извлекается из стека для выполнения МЦП прерванной программы. Система прерываний допускает одновременное поступление нескольких запросов разных уровней, а также нескольких запросов в пределах одного уровня. Очередность обслуживания нескольких запросов разных уровней решается аппаратным способом. Очередность обслуживания нескольких запросов в пределах одного уровня осуществляется программно.

Описанные режимы работы по запросам и прерыва-

ниям позволяют организовать взаимодействие между процессорными модулями в многопроцессорных системах.

Как видно из рис. 1.9, микроЭВМ СМ 1810 содержит два модуля центрального процессора: МЦП-16 и МЦП-1.

Модуль МЦП-16, структурная схема которого показана на рис. 1.10, построен на базе 16-разрядного микропроцессора К1810ВМ86. Роль многопроцессорной системной магистрали выполняет интерфейс И41. Локальная память модуля МЦП-16 емкостью 256 К байт имеет два входа: со стороны локальной магистрали микропроцессора и со стороны интерфейса И41. Такое архитектурное решение увеличивает гибкость в использовании модуля МЦП-16. В некоторых конфигурациях возможно использовать часть или всю локальную память, как системную, с возможностью доступа к ней от системных модулей. Для увеличения надежности локальная память защищена от случайных сбоев контролем по коду Хэмминга, который позволяет исключать одинарные ошибки и сигнализировать о двойных. В модуле МЦП-16 предусмотрена возможность установки арифметического сопроцессора типа 18087, который аппаратно реализует выполнение тригонометрических функций, десятичной арифметики и работу над числами с плавающей запятой. К местной шине МЦП-16 подсоединен программируемый последовательный интерфейс связи КР580ИК51А, реализующий интерфейс СТЫК С2, и программируемый параллельный интерфейс для периферийных устройств КР580ИК55, который реализует интерфейс радиальный параллельный ИРПР-М, описанный в кн. 5 настоящей серии. Интерфейс ИРПР-М используется в основном для связи с печатающими механизмами.

Для обеспечения совместимости микроЭВМ СМ 1800 и СМ 1810 в состав машины введен дополнительно 8-разрядный модуль МЦП-1, выполненный на базе микропроцессора КР580ИК80А. Краткое описание МЦП-1 приведено в [2]. В этом составе микроЭВМ СМ 1810 представляет собой инструментальную систему для подготовки программ реального времени, работающую под управлением операционной системы ДОС 1810.

Увеличение производительности и надежности СМ 1810 возможно путем построения многопроцессорных систем. Принципиально архитектура СМ 1810 позволяет обслуживать до 16 модулей центрального процессора типа МЦП-1 и МЦП-16, причем разнородные многопроцессорные системы, состоящие из МЦП-1 и МЦП-16, работают

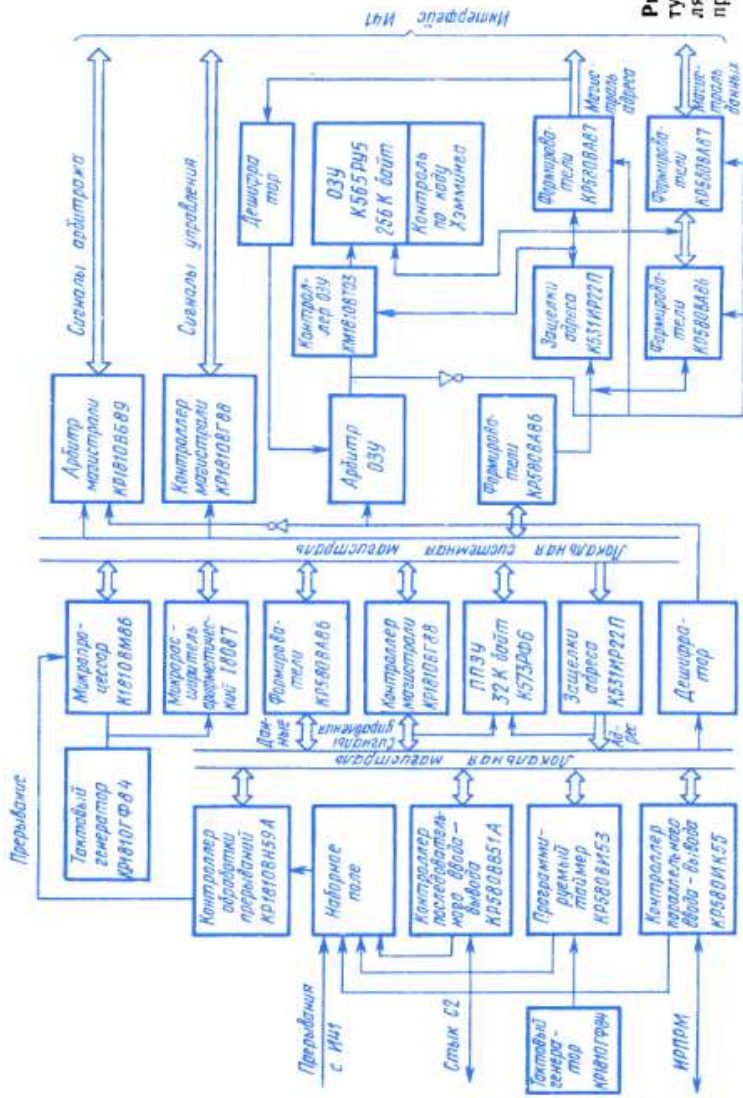


Рис. 1. 10. Структурная схема модуля центрального процессора МЦП-16

под управлением операционной системы ОС СФП 1810.

Программы, работающие под управлением ОС СФП 1810, налаживаются с помощью операционной системы ДОС 1810 и БОС 1810, причем БОС 1810 является инструментальным средством только для 16-рядных микропроцессорных систем.

Обмен информацией между модулями центральных процессоров в инструментальной системе СМ 1810 производится по следующему алгоритму.

Сообщение о конце передачи информации в модуль МЦП-1 формируется путем записи информации в младшую ячейку локальной оперативной памяти. При этом вырабатывается внутреннее прерывание, которое сообщает модулю МЦП-1 о конце передачи информации. Модуль МЦП-16 определяет конец передачи информации в локальную память двумя способами: 1) путем опроса соответствующей ячейки локальной памяти, в которую передающий процессор заносит признак конца сообщения (медленный способ); 2) по прерыванию, которое должен выставить передающий процессор.

Если мультимикропроцессорная система строится на модулях МЦП-16, то прерывания выдаются программно из передающего модуля. Если система строится на модулях МЦП-16 и МЦП-1, то прерывания из МЦП-1 выдаются программно с использованием одного из трех каналов модуля многорежимного таймера.

Прежде чем выдать прерывание, передающий модуль записывает в соответствующую ячейку памяти принимающего модуля признак конца сообщения. По прерыванию все модули процессоров переходят на анализ конца сообщения. Получит сообщение только тот модуль, который обнаружит в локальной памяти признак конца сообщения. После обнаружения конца сообщения этот признак программно сбрасывается в ноль.

Гибкая архитектура семейства микроЭВМ СМ 1800 обеспечена программным обеспечением, описание которого приводится в гл. 2.

1.3. Базовые модули СМ 1810

В составе СМ 1810 можно использовать большинство модулей из состава СМ 1800 [2]. Здесь приводится описание базовых модулей, которые обеспечивают макси-

мально возможную производительность микроЭВМ СМ 1810.

Модуль системного контроля МСК-16 СМ 1810.2005. Особое место в СМ 1810 занимает модуль системного контроля (МСК-16), который выполняется в виде одной платы и устанавливается в определенное место монтажного блока. Структурная схема МСК-16 приведена на рис. 1.11.

Модуль содержит узел нагрузочных сопротивлений линий интерфейса И41 и предназначен для выполнения функций арбитража запросов магистрали, обработки аварийных ситуаций в системе электропитания, сопряжения с органами управления на передней панели микроЭВМ СМ 1810 и выдачи в интерфейс сигналов синхронизации магистрали BCLK/ и CCLK/.

Генератор сигналов BCLK/ состоит из формирователя последовательности импульсов частотой 18 МГц, к выходу которого подключен триггер со счетным входом, формирующий последовательность импульсов частотой 9 МГц, которая через усилительный элемент подается в линию BCLK/ интерфейса И41.

В комплексах СМ 1810 предусмотрены три клавиши, позволяющие прервать исполнение текущей программы путем формирования сигналов прерывания. В узле связи с передней панелью размещены:

1) три схемы, предназначенные для приема сигналов от клавиш и формирования трех импульсов длительностью около 100 мс, которые через усилительные элементы подаются на выбранную линию запросов прерывания (INT0/ — INT7/);

2) схема формирования сигнала сброса, которая выдает в линию INIT/ интерфейса И41 сигнал сброса при

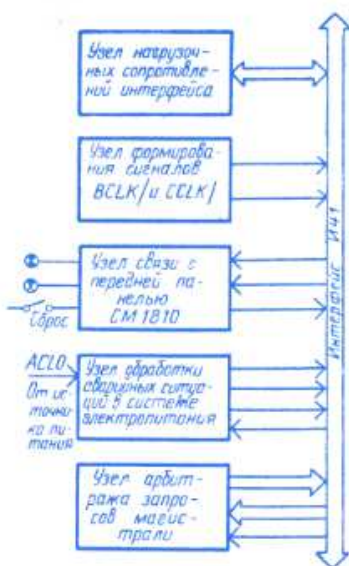


Рис. 1.11. Структурная схема модуля системного контроля МСК-16

нажатии клавиш СБРОС на панели управления комплекса СМ 1810 или при включении питания машины;

3) схема управления индикацией комплекса СМ 1810 предназначена для формирования сигнала, являющегося признаком исполнения процессором комплекса СМ 1810 команд программы, а также сигнала HALT, являющегося признаком останова процессора.

Узел обработки аварийных ситуаций в системе электропитания следит за снижением сетевого напряжения и выдает в процессор сигналы о неисправности системы электропитания.

Нагрузочные сопротивления предназначены для поддержания на линиях интерфейса И41 уровня логического 0 (высокий уровень напряжения) в том случае, когда все передатчики сигналов отключены от интерфейсных линий.

Генератор импульсов синхронизации CCLK/ формирует последовательность импульсов частотой 4 МГц, которые поступают на триггер, являющийся делителем частоты на 2. Через элемент усиления, реализованный на инверторе, серия импульсов частотой 2 МГц выдается на линию CCLK/ интерфейса И41.

Узел арбитража запросов к магистрали принимает от задатчиков (т. е. устройств, способных брать на себя управление интерфейсной магистралью, например процессоров, контроллеров, устройств прямого доступа в память), число которых может достигать 16, сигналы запроса BREQ0/ — BREQF/, выделяется наиболее приоритетный запрос и выдает в соответствующий задатчик сигнал разрешения доступа к интерфейсной магистрали (BPRN0/ — BPRNF/). Схема арбитража имеет два режима работы:

п а р а л л е л ь н ы й — приоритеты задатчиков неизменны, запрос BREQ0/ имеет высший приоритет, запрос BREQF/ — низший приоритет;

ц и к л и ч е с к и й — приоритеты задатчиков циклически изменяются, задатчик, завершивший обращение к шине интерфейса, приобретает низший приоритет, следующий задатчик — высший приоритет.

Выбор режима работы осуществляется распайкой переключки на модуле (отсутствие переключки соответствует параллельному режиму работы, а наличие ее — циклическому).

Модуль центрального процессора (МЦП-16) СМ 1810.2204. МЦП-16 построен на базе микропроцессора К1810БМ86 и предназначен для логической и ариф-

метической обработки информации и формирования интерфейса И41. Он характеризуется следующими основными техническими параметрами:

Система команд определяется архитектурой микропроцессора K1810VM86	
Разрядность данных	— 16
Система команд позволяет:	
организовать обработку двоичных шестнадцатеричных и десятичных данных (операции с фиксированной и плавающей запятой)	
осуществить операции с массивами переменной длины (поиск, пересылка) и операции с текстовой информацией	
Емкость внутренней постоянной памяти	— не менее 8 К байт
Емкость внутренней оперативной памяти	— 256 К байт
Максимальная емкость адресуемой памяти	— 16 М байт
Число адресуемых портов:	
ввода	— 65 536
вывода	— 65 536
Время выполнения команд — от 0,4 до 37,8 мкс, при этом время выполнения операций (над 16-разрядными числами со знаком):	
регистр—регистр	— 0,4 мкс
сложение регистр—регистр	— 0,6 мкс
умножение	— 30,6 мкс
деление	— 37,8 мкс
Число уровней прерывания	— 9
Модуль обеспечивает непосредственное подключение внешних устройств в соответствии с требованиями интерфейсов:	
— ИРПР-М	
— СТЫК С2	
(ГОСТ 18145—81)	

Структурная схема модуля центрального процессора МЦП-16 приведена на рис. 1.10. Центральным элементом его является микропроцессор K1810VM86, работающий с частотой 5 МГц. Микропроцессор осуществляет арифметические действия с 16-разрядными данными и содержит арифметическо-логическое устройство, четыре 16-битовых регистра-указателя базы и два 16-битовых индексных регистра. Система команд K1810VM86 включает в себя команды с переменным форматом, по которым выполняются 8- и 16-битовые операции со знаком и без знака над двоичными и двоично-десятичными операндами, а также над 1-байтовыми и 2-байтовыми строками. Более подробная информация о микропроцессоре K1810VM86 приводится в кн. 3 настоящей серии.

Обработка данных с плавающей запятой осуществляет-

ся микрорасширителем арифметическим I8087, который работает в режиме сопроцессора, т. е. при выполнении операции микрорасширителем I8087 микропроцессор K1810BM86 находится в состоянии ожидания конца операции.

Внутренняя синхронизация МЦП-16 обеспечивается тактовым генератором KP1810ГФ84, который генерирует синхросигналы частотой 5 МГц, сигнал сброса и готовности для микропроцессора K1810BM86 и сопроцессора I8087.

Архитектура МЦП-16 имеет иерархию из четырех магистралей: локальная системная, локальная, двухходовая для работы с внутренней оперативной памятью, многопроцессорная системная (интерфейс И41).

Производительность МЦП-16 непосредственно зависит от используемой магистрали, т. е. чем ближе к локальной магистрали выполняется команда, тем выше производительность.

Центральной для МЦП-16 является локальная системная магистраль, которая соединяет микропроцессор с локальной магистралью, двухходовой магистралью, интерфейсом И41 и ППЗУ. Максимальная производительность МЦП-16 достигается при работе с ППЗУ, так как обмен данными осуществляется непосредственно через локальную системную магистраль.

Локальная магистраль связана с локальной системной магистралью формирователя KP580BA86 и буферными регистрами K531ИР22, которые демultipлексируют магистраль адреса — данных микропроцессора. В качестве буферных регистров (защелки адреса) для демultipлексирования в МЦП-16 используются микросхемы K531ИР22, которые по сравнению с микросхемами KP580ИР82 (из серии K580, предназначенных для демultipлексирования шины адреса — данных) имеют большее быстродействие и меньшее потребление по питанию. Локальная магистраль — следующая по производительности за локальной системной, она предназначена для связи микропроцессора с контроллером обработки прерываний KP1810BH59A, контроллером последовательного ввода — вывода KP580BB51A (СТЫК С2), контроллером параллельного ввода — вывода KP580ИК55 (ИРПР-М) и программируемым таймером KP580ВИ53. Работа по локальной магистрали допускает выполнение операций внутри платы под непосредственным управлением от микропроцессора K1810BM86.

Следующая в иерархии — двухходовая магистраль для работы с ОЗУ, построенным на микросхемах памяти K565PY5. Структурная схема управления этой магистралью показана в правой части рис. 1.10. Как видно из рисунка, обмен информацией с ОЗУ возможен со стороны как локальной системной магистрали, так и многопроцессорной системной магистрали (интерфейс И41).

Функции распределения магистралей выполняет арбитр ОЗУ. Двухходовая магистраль ОЗУ может находиться в одном из трех состояний:

1 — локальная системная магистраль управляет двухходовой, но не использует ее (двухходовая магистраль ОЗУ не занята);

2 — локальная системная магистраль управляет двухходовой и использует ее (двухходовая магистраль ОЗУ занята);

3 — многопроцессорная системная магистраль (интерфейс И41) управляет двухходовой и использует ее (двухходовая магистраль ОЗУ занята).

Состояние 1 — холостое состояние двухходовой магистрали ОЗУ, которая управляется локальной с целью сведения к минимуму задержек при обращении к памяти микропроцессора K1810BM86. Как только микропроцессор обратится к ОЗУ модуля МЦП-16, схема управления двухходовой магистралью перейдет в состояние 2. Если магистраль занята, то локальная системная магистраль находится в состоянии ожидания до освобождения двухходовой. Таким образом обеспечивается минимальное время доступа к ОЗУ при обращении к нему от микропроцессора. После окончания операции чтения или записи в ОЗУ от микропроцессора двухходовая магистраль возвращается в состояние 1. На этом уровне работа с памятью осуществляется независимо от работы интерфейса И41, если при этом не требуется двухходовая магистраль для работы с ОЗУ модуля МЦП-16.

Если интерфейс И41 требует доступа к ОЗУ модуля МЦП-16, то двухходовая магистраль переходит из состояния 1 в состояние 3. В том случае, если двухходовая магистраль занята, интерфейс И41 ожидает ее перехода в состояние 3. Операция записи или чтения в состоянии 3 выполняется минимум за 150 нс, и по ее окончании двухходовая магистраль возвращается в состояние 1. Работа интерфейса И41 с двухходовой магистралью производится независимо от работы локальной магистрали с микропроцессором K1810BM86.

Связь с интерфейсом И41 микропроцессора К1810ВМ86 осуществляется при помощи арбитра КР1810ВБ89, который выдает и принимает сигналы арбитража доступа к интерфейсу И41, контроллера магистрали КР1810ВГ88, формирующего сигналы управления интерфейса И41, и формирователей КР580ВА87 сигналов адреса и данных.

В типичном применении модуль МЦП-16 является ведущим в микроЭВМ СМ 1810. Это означает, что он имеет возможность запрашивать и работать со всеми ресурсами, имеющимися в машине.

При пуске машины или включении питания тактовый генератор КР1810ГФ84 вырабатывает сигнал сброса (этот сигнал на рис. 1.10 не показан), который устанавливает в исходное состояние все микросхемы модуля МЦП-16. Обычно длительность этого сигнала составляет 100 мс. По окончании сигнала сброса начинается работа модуля по выполнению программы.

Модуль центрального процессора (МЦП-1) СМ 1800.2202. Чтобы обеспечить аппаратную и программную совместимость микроЭВМ СМ 1810 с микроЭВМ СМ 1800, в состав СМ 1810 вводится модуль МЦП-1, построенный на базе микропроцессора КР580ИК80А и предназначенный для логической и арифметической обработки 1-байтовых данных.

Модуль характеризуется следующими основными техническими параметрами:

Система команд определяется архитектурой микропроцессора КР580ИК80А

Разрядность данных — 8

Емкость локальной постоянной памяти — 8 К байт

Емкость локальной оперативной памяти — 8 К байт

Максимальная емкость адресуемой памяти — 1 М байт

Число адресуемых портов:
ввода — 256

вывода — 256

Время выполнения команд — от 2 до 8,5 мкс

Число уровней прерывания — 8

Локальная оперативная память — двухходовая, т. е. обращение к ней возможно как от микропроцессора, так и от устройств с прямым доступом, подключенных к интерфейсу И41

Модуль МЦП-1 эффективно используется в двухпроцессорной инструментальной системе, на которой готовят и налаживают программы для исполнительных систем, построенных на базе микропроцессора K1810BM86 и KP580IK80A.

Модуль контроллера накопителя на гибких магнитных дисках (КНГМД) СМ 1810.5125. Модуль КНГМД предназначен для управления накопителями на ГМД и совместно с ними образует устройство внешней памяти для микроЭВМ СМ 1810.

Модуль КНГМД может управлять четырьмя дисководами общей емкостью 4,8 М байт. При этом он имеет следующие технические характеристики:

- работает в любых системах, построенных на базе семейства микроЭВМ СМ 1800;
- управляет дисководами, работающими с гибкими дисками одинарной или двойной плотности, одно- или двусторонними,
- допускает возможность программирования пользователем параметров контроллера с целью подключения широкого ассортимента дисководов, работающих с дискеттами диаметром 133 мм или 208 мм;
- допускает возможность чтения и записи на одном или многих секторах длиной до 8192 байт;
- допускает адресацию до 16 М байт системной памяти;
- имеет питание + 5 В.

Контроллер НГМД построен на базе большой интегральной схемы БИС I8272, выполняющей функции управления дисководами гибких магнитных дисков. Интегральная схема I8272 может писать и читать один или несколько секторов, осуществлять циклический контроль считываемых данных и формировать код контроля CRC при записи данных. При каждом чтении или записи выбирается режим работы с одинарной или двойной плотностью, что обеспечивает стандартную запись одинарной плотности на нулевой дорожке, а затем переключение на запись с двойной плотностью на других дорожках.

Работа с дискеттой начинается по командам ввода — вывода от центрального процессора. Системная программа сначала инициализирует контроллер таким образом, чтобы он работал с конкретными дисководами и в заданном режиме. Затем производится форматирование дискетты под заданную операционную систему, т. е. дискетта делится на секторы той длины, с которыми работает в данный момент машина. Обмен данными между системной памятью и НГМД осуществляется при помощи БИС контроллера прямого доступа I8237A. Центральный процессор задает начальный адрес, длину массива дан-

ных и режим передачи для 18237А. По команде чтения или записи контроллер захватывает интерфейс И41 в монопольное владение и передает информацию. Монопольный захват интерфейса характеризуется тем, что до конца передачи информации ни одно из устройств, подсоединенных к интерфейсу, включая и центральный процессор, не может захватить интерфейс И41. Монопольный захват обеспечивает максимальную скорость передачи информации и исключает возможность ее потери.

Передача информации заканчивается после передачи заданного массива, о чем сообщается процессору сигналом прерывания. По сигналу прерывания центральный процессор считывает байт состояния контроллера, биты которого сообщают, насколько благополучно прошла передача информации. Во время записи информации вырабатывается код CRC, записываемый после заполнения каждого сектора. Обычно после записи проводится контрольное чтение. Если запись была ошибочна, то сектор, в котором была обнаружена ошибка, исключается из работы и делается попытка записи в другой сектор. Таким образом, возможна работа с дискеттами, которые имеют дефектные дорожки.

Модуль КНГМД имеет программные драйверы в составе всех операционных систем, поставляемых для микроЭВМ СМ 1810.

Модуль контроллера накопителя на магнитных дисках СМ 1810.5123. Модуль КНМД предназначен для управления накопителями на жестких магнитных дисках большой емкости, имеющих интерфейс SMD, и совместно с ними образует устройство внешней памяти для микроЭВМ СМ 1810. Модуль КНГМД может управлять четырьмя магнитными дисками общей емкостью до 2,4 Г байт.

Интерфейс SMD является международным стандартом для накопителей большой емкости. Основные достоинства интерфейса SMD — высокая скорость и надежность передачи данных. Скорость передачи — 1,2 М байт/с — достигается за счет разделения линий данных для каждого накопителя. Управляющие сигналы для накопителей общие и идут последовательно от одного накопителя к другому.

Модуль КНМД имеет следующие технические характеристики:

— модуль может работать в любых системах, построенных на базе семейства микроЭВМ СМ 1800;

— модуль управляет четырьмя накопителями, имеющими интерфейс SMD;

— модуль обеспечивает: управление накопителями емкостью от 12 М байт до 2,4 Г байт; обнаружение и коррекцию ошибок данных; полную буферизацию сектора данных;

— возможность адресации — до 16 М байт системной памяти, возможность программирования пользователем параметров контроллера, что допускает подключение широкого ассортимента накопителей, имеющих интерфейс SMD.

Модуль КНМД построен на базе процессора ввода — вывода КМ1810ВМ89, обеспечивающего работу по двум каналам прямого доступа и управление накопителями по программе, записанной в нестираемую память.

Программа реализует так называемый метод связанного списка, позволяющий пользователю выполнять множество различных дисковых операций.

Обмен информацией между накопителем и системной памятью осуществляется через буфер сектора, который предотвращает ошибки пропуска данных и позволяет занимать модулю в системе не самое приоритетное место. Высокая надежность записи и считывания данных обеспечивается использованием корректирующего кода. При записи сектора или массива данных генерируется 32-битовый корректирующий код, который добавляется к записанному массиву. Во время чтения та же логика формирует 32-битовый корректирующий код и сравнивает его с записанным.

Если коды сравнимы, то данные достоверны. Если коды не сравнимы, то начинает работать программа исправления ошибки. Ошибка может быть исправлена, если ошибочный всплеск не превышал длину в 11 бит. Все проверки осуществляются микропроцессором КМ1810ВМ89 и совершенно незаметны для пользователя. По окончании передачи выдается сигнал прерывания, по которому центральный процессор считывает байт состояния контроллера с сообщением, насколько благополучно прошла передача информации.

Во время форматирования накопителя центральный процессор проверяет целостность дорожек на диске. Если дорожка дефектная, то процессор присваивает ей код дефектной дорожки и вводит адрес запасной дорожки. Если контроллер обращается к дорожке, предварительно помеченной как дефектная, то он автоматически находит ее и обращается к приписанной запасной дорожке. Работа с запасной дорожкой абсолютно незаметна для пользователя.

Модуль КНМД имеет программные драйверы в составе всех операционных систем, поставляемых для микроЭВМ СМ 1810.

Модуль контроллера накопителя на жестких и гибких магнитных дисках (КНМГМД) СМ 1810.5126. Предназначен для управления накопителями. Фактически модуль состоит из двух контроллеров: контроллера жесткого диска типа «Винчестер», имеющего интерфейс ST506, и контроллера гибкого диска. Интерфейс ST506 является международным стандартом для накопителей средней емкости типа «Винчестер».

Общей частью для обоих контроллеров является схема, обеспечивающая связь с интерфейсом И41. Модуль может управлять двумя накопителями типа «Винчестер» и четырьмя накопителями на ГМД диаметром 133 мм или 208 мм.

Общие технические характеристики модуля КНМГМД:

— модуль работает в любой системе, построенной на базе семейства микроЭВМ СМ 1800;

— модуль управляет двумя накопителями типа «Винчестер», имеющими интерфейс ST506, и четырьмя накопителями на ГМД одинарной или двойной плотности, односторонних или двусторонних диаметром 208 или 133 мм;

— модуль обеспечивает: реализацию международного интерфейса ST506, обнаружение и коррекцию ошибки в данных, полную буферизацию сектора данных, возможность адресации до 16 М байт системной памяти, возможность программирования пользователем параметров контроллера, что допускает подключение широкого ассортимента накопителей на гибких и жестких магнитных дисках.

Модуль дает возможность пользователю построить систему внешней памяти средней емкости минимальной стоимости. Такие системы особенно популярны среди системных программистов, которым требуется один гибкий магнитный диск для ввода программы и быстродействующий жесткий диск типа «Винчестер» емкостью 10—20 М байт для эффективной трансляции и редактирования программ.

Подобно модулю КНМД, в модуле КНМГМД управляет жесткими дисками микропроцессор КМ1810ВМ89 по аналогичному алгоритму.

Управление гибким диском осуществляется контроллером I8272 по тому же алгоритму, по которому выполняется управление в модуле КНГМД.

Обмен данными между системной памятью и модулем КНМГМД осуществляется двумя каналами прямого

доступа в контроллере ввода — вывода КМ1810ВМ89, так же как и в модуле КНМД.

Модуль КНМГМД имеет программные драйверы в составе всех операционных систем, поставляемых для микроЭВМ СМ 1810.

Модуль оперативный запоминающий МОЗ 256 СМ 1810.3515. Он предназначен для приема, хранения и выдачи оперативной информации в качестве встроенной оперативной памяти в составе микроЭВМ СМ 1810. Модуль имеет следующие технические характеристики:

Объем	— 256 К байт
Разрядность	— 8 и 16 бит
Порядок обращения	— произвольный
Выполняемые операции	— запись слова (ЗПС), чтение слова (ЧТС), запись байта (ЗПБ), чтение байта (ЧТБ)
Цикл обращения	— при операциях ЧТС, ЗПС, ЧТБ не более 0,7 мкс; при операции ЗПБ не более 1,4 мкс
Время выборки	— не более 0,65 мкс

Модуль обеспечивает коррекцию одинарной и обнаружение двойной ошибки.

Структурная схема модуля приведена на рис. 1.12.

Узел приема осуществляет формирование адреса обращения к требуемой ячейке памяти при обращении к модулю со стороны интерфейса И41.

Узел обработки данных осуществляет прием и выдачу данных на (из) интерфейс(а) И41. В его состав входит корректор, обеспечивающий при операциях записи фор-

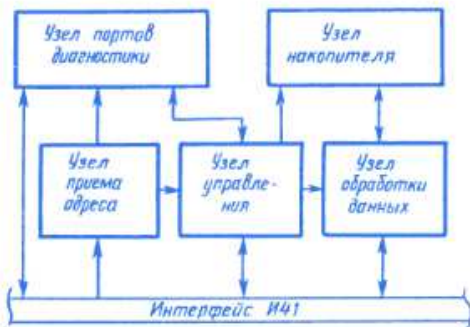


Рис. 1.12. Структурная схема модуля МОЗ 256 СМ 1810.3515

мирование контрольных разрядов накопителя. При операциях чтения корректор формирует признаки одинарной и двойной ошибок и в случае одинарной ошибки производит коррекцию данных и выдачу их через соответствующие буферы на интерфейс И41.

Узел управления формирует сигналы управления остальными узлами модуля и соответствующую временную диаграмму. В его состав входит контроллер памяти КМ1810ВТ03, осуществляющий формирование управляющих сигналов для динамических микросхем памяти, прием и мультиплексирование адресов строки и столбца, а также обеспечивает формирование режима регенерации.

Узел накопителя предназначен для записи, хранения и выдачи информации и представляет собой матрицу микросхем памяти К565РУ5 (64К×1). Матрица содержит два ряда по 22 микросхемы К565РУ5. Разряд данных включает в себя по одной микросхеме из каждого ряда; таким образом, в матрице всего 16 информационных и 6 контрольных разрядов. Полная емкость накопителя 128К×22 бит, где К=1024 бит.

Узел портов диагностики осуществляет прием и выдачу информации о диагностике модуля и состоит из портов ввода — вывода, в которых хранится информация о работоспособности модуля.

Модуль оперативный запоминающий МОЗ 4М СМ 1810.3516. В состав СМ 1810 входит еще один модуль оперативной памяти МОЗ 4М, который отличается от МОЗ 256 большой емкостью (до 4 М байт).

Остальные параметры МОЗ 4М аналогичны МОЗ 256.

МОЗ 4М состоит из пяти плат, которые устанавливаются в определенное место СМ 1810.40 или СМ 1810.41. Из них одна плата выполняет функции контроллера памяти, остальные четыре платы — функции накопителя. Платы накопителя полностью взаимозаменяемы и служат для наращивания накопителя блоками по 1 М байт до 4 М байт. Минимальная емкость МОЗ 4М — 1 М байт.

В комплексах СМ 1810.50 и СМ 1810.51 модуль МОЗ 4М имеет емкость 1М байт и состоит из двух плат (одна плата — контроллер, другая — накопитель на 1 М байт). Следует помнить, что МОЗ 4М имеет ограниченное применение только в комплексах СМ 1810.40, СМ 1810.50, СМ 1810.41 и СМ 1810.51.

1.4. Конструкция и варианты конструктивного исполнения микроЭВМ

Конструкция семейства микроЭВМ СМ 1800 представляет собой набор различных конструктивных компонентов, обеспечивающих широкий диапазон системных применений, а также большие возможности сервиса при эксплуатации систем, построенных на базе данного семейства микроЭВМ.

Основной конструктивный компонент — блок элементов (БЭ) — представляет собой печатную плату размера $233,35 \times 220$ мм с размещенными на ней радиодеталями, двумя или тремя вилками разъема СМП59 и планкой с экстракторами для извлечения, установки и фиксации блока элементов в монтажном блоке. Из блоков элементов собираются вычислительные системы на базе семейства микроЭВМ СМ 1800. Он может вставляться в монтажные блоки микроЭВМ как СМ 1800, так и СМ 1810.

Конструкция микроЭВМ СМ 1800 описана в [2]. Основными частями ее являются:

- монтажный блок, рассчитанный на установку до 10 блоков элементов, размещаемых в нем горизонтально. Блоки элементов вставляются в гнезда разъемов СМП59, установленных на монтажной панели и объединенных интерфейсом И41. На боковой поверхности блока укреплены два вентилятора;

- автономный комплектный блок, рассчитанный на установку двух монтажных блоков и блока питания;

- тумба (размера $725 \times 600 \times 800$ мм) и стойка (размера $1800 \times 600 \times 800$ мм), являющиеся несущими конструкциями и предназначенные для произвольного размещения, конструктивно-художественного объединения и механической защиты автономных комплектных блоков.

Конструкция микроЭВМ СМ 1810 (по сравнению с СМ 1800) претерпела изменения в отношении снижения габаритов и металлоемкости. Было решено отказаться от автономного комплектного блока; кассеты (в СМ 1800 монтажные блоки) размещать непосредственно в тумбе, стойке или приборе, а источники питания — непосредственно в кассете на разъемах СМП59. Кроме того, для улучшения обдува блоков элементов было решено размещать их в кассете вертикально. Таким образом, основными конструктивными компонентами микроЭВМ СМ 1810 являются: кассета; прибор размера $482 \times 400 \times$

×320 мм; тумба размера 540×400×600 мм; стойка размера 540×400×1200 мм.

В составе СМ 1810 имеется шесть типов кассет:

— на 9 блоков элементов с разводкой интерфейса И41 и местами для установки одного блока питания, накопителя на ГМД диаметром 130 мм и НМД типа «Винчестер» диаметром 130 мм. Шаг между БЭ — 17,5 мм;

— на 15 блоков элементов с разводкой интерфейса И41 и местами для установки двух блоков питания;

— на 10 блоков элементов с разводкой интерфейса И41, 5 блоков элементов для установки модуля оперативной памяти емкостью до 4 М байт и местами для установки двух источников питания.

В эти кассеты устанавливаются модуль системного контроля МСК-16 и ведущий модуль центрального процессора.

Следующие три типа кассет служат в качестве расширителей интерфейса И41 и для увеличения числа блоков элементов в системе:

— на 15 блоков элементов с разводкой интерфейса И41, модуль расширения МР1 и двумя блоками питания;

— на 19 блоков элементов с разводкой интерфейса И41 только для устройств ввода — вывода, модулем расширения МР2 и одним блоком питания;

— на 16 блоков элементов с разводкой интерфейса И41 только для устройств ввода — вывода, модулем расширения МР2 и одним блоком питания. Эта кассета предназначена для установки модулей, имеющих компоненты, превышающие 17 мм. Поэтому шаг между блоками элементов здесь 20 мм, в то время как в остальных кассетах 17,5 мм.

В кассетах используются два типа модулей расширения: МР1 и МР2. Модуль расширения МР1 позволяет подключать к системе модули, работающие по прямому доступу в память; модуль расширения МР2 — только те модули, которые не используют возможность обращения по прямому доступу к памяти.

Прибор представляет собой коробку, в которую устанавливается кассета на девять посадочных мест для создания настольной микроЭВМ.

Тумба предназначена для установки в нее до двух кассет любого типа для получения вычислительного комплекса средней мощности, который можно убрать под канцелярский стол.

Стойка предназначена для установки в нее до трех

кассет любого типа с целью создания вычислительного комплекса необходимой мощности.

Блоки расширения позволяют создавать системы из нескольких стоек.

Еще одна конструктивная единица — кроссовый блок, предназначенный для подсоединения сигналов с датчиков к модулям устройств связи с объектом. Он представляет собой конструкцию, устанавливаемую в тумбу или стойку так же, как кассеты. В кроссовом блоке устанавливаются колодки для подсоединения проводов, модули аналогового питания и блоки элементов размера 160××100 мм. Максимальное число колодок в кроссовом блоке — 18 на 36 проводов каждая. Каждый блок аналогового питания занимает место четырех колодок. Каждый блок элементов занимает место одной колодки. Сочетание числа колодок, модулей аналогового питания и блоков элементов в кроссовом блоке произвольно. Агрегатно-модульная конструктивная реализация микро-ЭВМ СМ 1810 позволяет достаточно просто проектировать комплексы минимальной стоимости для конкретного пользователя.

Программное обеспечение семейства микроЭВМ СМ 1800



Программное обеспечение СМ 1800 и СМ 1810 включает в себя следующие группы средств: операционные системы общего назначения; инструментальные операционные системы; операционные системы реального времени; базовое прикладное программное обеспечение.

На рис. 2.1 приведено развитие операционных систем СМ 1800 и СМ 1810.

Операционные системы общего назначения используются в непромышленной сфере для организации работы пакетов прикладных программ, а также как операционная среда для профессиональных персональных применений. Они включают в себя довольно простой набор сервисных программ, под управлением операционных систем общего назначения работает большое число трансляторов с различных языков программирования. В число операционных систем общего назначения для СМ 1800 входят: система программного обеспечения (СПО); операционная система ОС1800; мобильная операционная система МикроДОС.

Дальнейшее развитие операционные системы общего назначения получили в микроЭВМ СМ 1810, в состав которой входят операционные системы Микрос-86; малая дисковая операционная система МДОС1810.

Операционные системы данного класса получили наибольшее распространение. Так, во всем мире фактически общепринятыми и стандартными для 8-разрядных микроЭВМ стала операционная система CP/M-80 (с ней совместимы ОС1800 и МикроДОС); для 16-разрядных микроЭВМ операционной системы MSDOS [3] (с ней совместима МДОС1810).

Инструментальные операционные системы предназначены для разработки сложных программных проектов, в первую очередь систем реального времени. Отсюда следует и имеющееся в них значительное количество сервисных программ (утилит), в том числе ориентированных на гибкую организацию пакетного режима и сопровождение программных продуктов. В инструментальных системах большое внимание уделяется ориентации на модульное программирование. Обязательным стало требование совместимости объектов файлов, генерируемых различными компиляторами, что позволяет разрабатывать прикладные системы на нескольких языках программирования. Хотя по реализации инструментальные системы аналогичны операционным системам общего назначения, область применения их имеет существенные особенности.

В число инструментальных систем входят две совместимые дис-

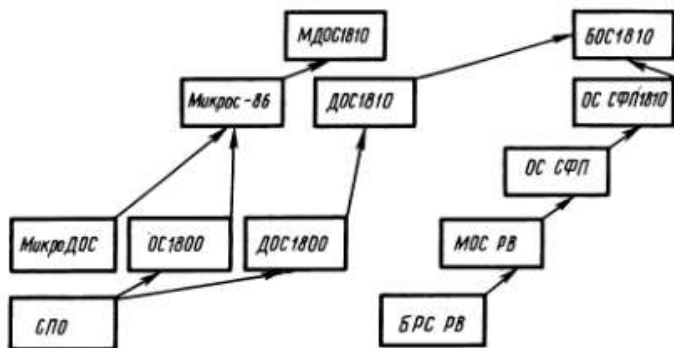


Рис. 2.1. Развитие операционных систем СМ 1800 и СМ 1810

ковые операционные системы: для СМ 1800 (ДЭС1800) и СМ 1810 (ДЭС1810).

К числу чисто исполнительных операционных систем реального времени для СМ 1800 относятся:

- базовая резидентная система реального времени (БРС РВ);
- мультипрограммная операционная система реального времени (МОС РВ);
- мультимикропроцессорная операционная система реального времени со специализацией функций процессоров (ОС СФП).

Все эти системы совместимы между собой, а также с распространенной операционной системой реального времени RMX/80 [4]. Таким образом, пользователь может выбирать операционную среду — от простой системы БРС РВ до сложной системы ОС СФП.

Архитектура рассматриваемых операционных систем отражает современные достижения теории организации мультипрограммных систем, максимальную унификацию способов обращения к компонентам операционных систем.

Разработанная для СМ 1810 система ОС СФП 1810 функционально идентична системе МОС РВ и ОС СФП для СМ 1800. Учитывая широкое использование в прикладных системах языка ПЛ/М, можно перенести созданные для микроЭВМ СМ 1800 прикладные системы реального времени на 16-разрядную микроЭВМ СМ 1810 путем перетрансляции исходных текстов и перекомпоновки системы с включением других наборов библиотек, входящих в состав операционной системы.

Принципиально новая система — большая операционная система БОС1810, совмещающая исполнительный и инструментальный режимы работы. БОС1810 включает те же языки программирования для 16-разрядного микропроцессора, что и ДЭС1810. Она имеет объектно-ориентированную архитектуру, позволяющую расширять систему новыми объектами (наряду с существующими) и вводить новые операции для работы с ними. Иерархическая структура БОС1810 включает в себя ядро, базовую систему ввода — вывода, расширенную систему ввода — вывода, интерфейс с оператором, универсальный программный интерфейс, драйвер терминала, отладчик, загрузчик и другие компоненты-слон.

В операционных системах СМ 1810 применяется универсальный программный интерфейс (УПИ) как средство доступа прикладных

программ к файлам. Набор системных вызовов УПИ прост и достаточен для выполнения основных функций, требуемых для решения задач пользователя.

УПИ повышает степень переносимости программ СМ 1810 из одной операционной среды в другую. Более подробно характеристики УПИ рассматриваются в приложении.

Базовое прикладное программное обеспечение СМ 1800 и СМ 1810 включает в себя программные средства следующих основных типов:

- базы данных;
- текстовые редакторы, процессоры и форматоры;
- базовые графические средства, в том числе программное обеспечение для деловой графики;
- средства межмашинного обмена, эмуляции терминалов верхней ЭВМ и программное обеспечение для построения локальных сетей ЭВМ;
- интегрированные пакеты (базы данных, текстообработка, генерация и обработка таблиц, машинная графика и межмашинный обмен информацией).

В § 2.4 рассматриваются основные прикладные программы для СМ 1800, разработка которых уже завершена. Прикладное программное обеспечение в настоящее время бурно развивается, многие прикладные программы представляют собой готовые к применению высокоэффективные проблемно-ориентированные системы.

2.1. Операционные системы общего назначения

Операционные системы общего назначения для микро-ЭВМ получили наибольшее распространение среди всех операционных систем. Под их управлением работает большое число трансляторов, пакетов прикладных программ и других программных средств.

Система программного обеспечения (СПО). Это наиболее простая однопользовательская однопрограммная система [5]. В ее состав входят монитор-отладчик MONID, сервисные программы (утилиты), редактор текстов, ассемблер, упрощенная версия языка ПЛ/М и интерпретатор минимального языка БЭЙСИК. Утилиты СПО обеспечивают выполнение основных функций в процессе разработки программного обеспечения, т. е. создание, переименование и удаление файлов на ГМД в качестве внешней памяти, форматирование (инициализация) ГМД и т. д. Эта система ориентирована на пользователей, не являющихся профессиональными программистами, и позволяет эффективно поддерживать процесс создания и отладки относительно небольших программ.

Ассемблер и ПЛ/М генерируют при трансляции абсолютные загрузочные модули. Это, естественно, упрощает процесс трансляции, повышает его скорость, однако не дает возможности применять методы модульного программирования.

В СПО используется два типа логической организации дискетт:

— дискетты программ содержат готовые к исполнению отлаженные программы, в том числе утилиты и трансляторы СПО (вся СПО как простейшая система помещается на одной дискетте программ);

— дискетты данных содержат создаваемую пользователем информацию. Например, исходные тексты пользовательских программ, оттранслированные абсолютные объектные модули и т. д. Редактор текстов (программа RED) позволяет создавать и корректировать текстовые файлы на дискеттах данных. Ассемблер и ПЛ/М в качестве входной информации воспринимают содержимое какого-либо файла на дискетте данных. В результате трансляции в файле на дискетте данных также образуется абсолютный модуль в HEX-формате. Абсолютный модуль загружается для исполнения в память СМ 1800 с помощью загрузчика (программы LOAD).

Отладка в СПО осуществляется монитором-отладчиком MONID, который помимо интерфейсных подпрограмм содержит набор основных отладочных команд для выполнения процесса отладки в диалоговом режиме. После отладки пользовательская программа помещается на дискетту программ специальной утилитой WD.

Операционные системы общего назначения ОС1800 и МикроДОС. Для СМ 1800 имеется две более сложных операционных системы общего назначения: операционная система ОС1800 [5, 6]; мобильная операционная система для микроЭВМ МикроДОС [7].

Обе системы совместимы друг с другом, а также с СР/М-80 — наиболее широко распространенной в мире системой для 8-разрядных микроЭВМ. Для этих систем имеется большое число готовых пакетов прикладных программ, включая базовые, часть из которых кратко рассматривается далее.

Операционные системы общего назначения состоят из двух систем:

1) базовой дисковой операционной системы (БДОС), осуществляющей логический ввод — вывод и взаимодействие пользователя с микроЭВМ;

2) базовой системы ввода — вывода (БСВВ), в которой реализуется физический ввод — вывод для основных устройств микроЭВМ.

Кроме того, они содержат процессор консольных команд (ПКК) с некоторым набором системных утилит,

называемых резидентными. Для прочих утилит, трансляторов и пакетов выделяется так называемая область транзитных программ (ОТП). Загружаемые в ОТП утилиты называются транзитными.

Работы с файлами — одна из основных функций операционных систем общего назначения. Для задания полного имени файла в этих системах используется синтаксис:

[d:]filename[.typ]

где d — имя диска; filename — имя файла (до 8 знаков); typ — тип файла или расширение имени файла (до 3 знаков).

Файлы делятся на файлы данных (в том числе содержащие исходные тексты программ), объектные и загрузочные (исполняемые).

По отношению к файловой системе каждый диск содержит каталог и данные (файлы). Распределение дискового пространства — автоматическое. По отношению к операционной системе диск скорее логическое, чем физическое понятие, объем которого может быть больше емкости физического тома (носителя). Записи в файлах имеют длину 128 байт. Пространство на диске выделяется блоками, каждый из которых содержит некоторое число записей, кратное 8.

Входящие в состав операционных систем общего назначения утилиты позволяют: выводить каталог; удалять файл или группу файлов; копировать файлы; генерировать носитель операционной системы; организовывать работу в пакетном режиме; сравнивать файлы; осуществлять отладку программ пользователя, включая режим трассировки.

Кроме того, в МикроДОС имеются утилиты для отображения файла, установки атрибутов, доступа к физическому содержимому дисков, дизассемблер (REMAC); в ОС1800 — утилиты для преобразования файлов СПО в файлы ОС1800, HEX-формата в формат загрузочных модулей (LOAD).

В операционной системе МикроДОС имеются интерпретаторы минимального языка БЭЙСИК и БЭЙСИК/F, компиляторы БЭЙСИК/F, "С" и ПАСКАЛЬ, а также макроассемблер. В ОС1800 также работают языки макроассемблер, ФОРТРАН, ПАСКАЛЬ, интерпретатор и компилятор языка БЭЙСИК. Кроме того, в ОС1800 имеются компиляторы МИБОЛ (подмножество языка

КОБОЛ), ПЛ/1 и генерирующий абсолютные объектные модули упрощенный компилятор ПЛ/М.

Для обслуживания перемещаемых объектных модулей имеются средства компоновки и библиотекари объектных модулей.

В рассматриваемых операционных системах выходом (результатом работы) большинства трансляторов являются перемещаемые объектные модули, что позволяет применять методы модульного программирования. Уровень языков, состав утилит позволяют с помощью операционных систем общего назначения разрабатывать на СМ 1800 средние программные проекты, состоящие из многих модулей. Для разработки более сложных программных комплексов рекомендуется использовать инструментальную операционную систему ДОС1800, которая рассматривается далее.

Операционная система Микрос-86. Микрос-86 является развитием ОС1800 для 16-разрядной микроЭВМ СМ 1810. Файловая структура Микрос-86 совместима с принятой в ОС1800 и МикроДОС. Набор системных вызовов приближен, насколько это возможно, к набору системных вызовов 8-разрядных операционных систем общего назначения. Это позволяет достаточно легко конвертировать прикладные программы для их исполнения под управлением Микрос-86.

Далее рассматриваются отдельные свойства и характеристики Микрос-86.

Команды. Микрос-86 выполняет отдельные функции, вызываемые соответствующей командой, вводимой с клавиатуры. Командная строка вводится оператором после системной подсказки, например

A>

где А — имя дискового устройства, используемого по умолчанию, т. е. в тех случаях, когда какое-либо имя не указывается в явном виде.

При вводе команд возможно их редактирование. Вводимая командная строка состоит из команды и необязательных параметров. Существует два вида команд: встроенные (резидентные) и транзитные. Встроенные команды находятся в оперативной памяти СМ 1810 как часть операционной системы, они могут исполняться сразу же после их вызова оператором. Транзитные команды хранятся как файлы на диске, они загружаются в память. Имена файлов, выполняющих транзитные

команды, соответствуют именам команд, но имеют тип CMD.

В ряде программ (команд) Микрос-86 используются имена файлов с предопределенными типами:

- CMD — загружаемая и исполняемая программа;
- OBJ — временный файл;
- H86 — исполняемый файл в HEX-формате;
- SUB — командный файл, исполняемый в пакетном режиме с помощью команды SUBMIT.

Так же как в 8-разрядных операционных системах, здесь возможно использование специальных знаков ? и * при описании группы файлов в именах и типах файлов.

К числу встроенных команд Микрос-86 относятся:

- DIR — вывод списка файлов на диске, имеющих атрибут D;
- DIRS — вывод списка файлов на диске, имеющих атрибут S;
- ERA — исключение (уничтожение) файла;
- REN — переименование файла;
- TYPE — отображение текстового файла на экране консоли;
- USER — изменение номера пользователя.

К числу важнейших транзитных команд относятся:

- COPYDISK — копирование диска;
- GENCMD — использование выхода ассемблера для генерации загружаемого и исполняемого файла;
- PIP — объединение и копирование файлов;
- STAT — отображение информации о состоянии системы, а также изменение атрибутов файлов и устройств как некоторых параметров;
- SUBMIT — организация исполнения в пакетном режиме;
- TOD — установка и отображение системной даты и времени.

Кроме указанных команд в Микрос-86 имеются отладчик и редактор текстов. В число языков программирования Микрос-86 помимо ассемблера входят БЭЙСИК, ФОРТРАН, ПАСКАЛЬ и "С"

Системные вызовы. Системные вызовы Микрос-86 во многом совпадают с системными вызовами операционных систем ОС1800 и микроДОС. Набор системных вызовов приведен ниже, причем * отмечены новые или измененные (по отношению к ОС1800 и микроДОС) вызовы:

- | | |
|---|--|
| 0* — Сброс системы | 27* — Получение вектора распределения памяти |
| 1 — Ввод с консоли | 28 — Установ защиты записи |
| 2 — Вывод на консоль | 29 — Получение вектора защиты записи |
| 3 — Ввод из устройства ввода (перфоленты) | 30 — Установ атрибутов файла |
| 4 — Вывод на устройство вывода (перфоленту) | 31* — Получение адреса блоков параметров диска |
| 5 — Вывод на печать | 32 — Получение — установка кода пользователя |
| 6* — Прямой обмен с консолью | 33 — Произвольное чтение |
| 7* — Чтение байта подключения устройств | 34 — Произвольная запись |
| 8* — Запись байта подключения устройств | 35 — Вычисление размера файла |
| 9 — Вывод строки на консоль | 36 — Установ номера произвольной записи |
| 10 — Чтение строки с консоли | 37* — Сброс драйвера (диска) |
| 11 — Получение состояния консоли | 40 — Произвольная запись нулями |
| 12 — Получение версии системы | 50* — Прямой вызов БСВВ |
| 13 — Сброс дисковой системы | 51* — Установка сегмента прямого доступа |
| 14 — Выбор диска | 52* — Получение сегмента прямого доступа |
| 15 — Открытие файла | 53* — Получение максимальной памяти |
| 16 — Закрытие файла | 54* — То же, что 53, но в диапазоне памяти |
| 17 — Поиск файла в оглавлении | 55* — Выделение памяти |
| 18 — Продолжение поиска файла в оглавлении | 56* — Выделение абсолютной памяти |
| 19 — Исключение файла | 57* — Освобождение памяти |
| 20 — Последовательное чтение | 58* — Освобождение всей памяти |
| 21 — Создание файла | 59* — Загрузка программы |
| 22 — Последовательная запись | |
| 23 — Переименование файла | |
| 24 — Получение вектора состояния дисков | |
| 25 — Получение текущего диска | |
| 26 — Установ адреса прямого доступа | |

Системный вызов 50 позволяет обратиться непосредственно к БСВВ, среди которых доступны следующие подпрограммы:

- | | |
|--|---|
| 0 — Переход на «горячий» старт | 10 — Установ номера дорожки |
| 1 — Переход на старт | 11 — Установ номера сектора |
| 2 — Проверка готовности ввода клавиатуры консоли | 12 — Установ смещения адреса буфера для прямого доступа |
| 3 — Чтение знака с консоли | 13 — Чтение требуемого сектора |
| 4 — Запись знака на консоль | 14 — Запись требуемого сектора |
| 5 — Печать знака | 15 — Получение состояния печати |
| 6 — Запись знака на устройство вывода | 16 — Преобразование логического номера сектора в физический |
| 7 — Чтение знака из устройства ввода | 17 — Установ адреса сегмента для чтения |
| 8 — Перемещение на нулевую дорожку | 18 — Получение смещения для |
| 9 — Выбор драйвера (диска) | |

При обращении к БСВВ передача номера требуемого действия осуществляется через регистр CL микропроцессора K1810BM86.

Малая дисковая операционная система (МДОС1810). Она совместима с распространенной операционной системой для 16-разрядных микроЭВМ MSDOS. Под ее управлением работает значительная часть пакетов программ, предназначенных для персональной ЭВМ типа IBM PC. Структурно МДОС1810 состоит из четырех частей:

- начального загрузчика;
- базовой системы ввода — вывода, осуществляющей управление устройствами ввода — вывода;
- системы управления файлами;
- процессора команд.

Имена файлов имеют длину 1—8 знаков, за которыми может следовать (после точки) расширение имени (1—3 знака). Имена файлов, а также информация об их размерах, размещении на дисках, дате и времени создания хранятся в оглавлении дисков. Таблица распределения файлов на диске определяет местоположение файлов и свободное дисковое пространство. Для повышения надежности на диске хранятся две копии таблицы распределения файлов.

Файлы на дисках организуются по иерархическому принципу, т. е. имеются вложенные директории. Имеется возможность при запуске системы автоматически запустить программы с помощью командного файла AUTOEXEC. BAT. Как и в Микрос-86, ввод команд осуществляется с помощью процессора команд. Команды, включаемые в процессор команд, являются резидентными, т. е. они исполняются сразу же после вызова. Команды, размещаемые на дисках в файлах и считываемые с диска перед запуском, являются транзитными.

Основные команды МДОС1810 следующие:

- ASSIGN — переназначение дисковых устройств;
- ATTRIB — определение или отображение атрибута файла;
- BREAK — установ режима обработки прерываний от оператора;
- CHDIR — изменение текущей директории на диске;
- COMP — сравнение файлов;

- COPY — копирование файлов (возможно объединение файлов);
- CTTY — изменение назначения консоли с первичной на вторичную;
- DATE — ввод даты;
- DEL и ERASE — удаление файлов;
- DIR — вывод оглавления диска (директории);
- DISKCOMP — сравнение дисков;
- DISKCOPY — копирование дисков;
- EXE2BIN — преобладание файлов с расширением. EXE в форму, совместимую с программами, имеющими расширение .COM;
- FIND — нахождение всех строк в файле, содержащих требуемую текстовую последовательность;
- FORMAT — форматирование дисков;
- LABEL — установ имени диска;
- MKDIR — создание директории;
- MODE — определение режима асинхронного адаптера или назначение на него печатающего устройства;
- MORE — страничный вывод файла на экран;
- PATH — установ директории для поиска файлов с программой (например, команд МДОС1810);
- PRINT — организация печати в фоновом режиме;
- PROMT — выдача имен текста системного запроса (подсказки) процессором команд;
- RECOVER — восстановление файлов, содержащих дефектные (испорченные) данные;
- RENAME — переименование файлов;
- RMDIR — удаление директорий;
- SHARE — установ доступа к файлам;
- SORT — сортировка данных в файле;
- SYS — копирование системных файлов;
- TIME — ввод времени;
- TREE — вывод всех вложенных директорий для определенного устройства;
- TYPE — вывод файла;
- VER — вывод версии МДОС1810;
- VERIFY — установ — отмена режима записи на диск с проверкой;
- VOL — отображение имени диска.

Отличительной особенностью системы МДОС1810 являются так называемые стандартные ввод и вывод. Во время запуска системы стандартный ввод назначается на клавиатуру, а стандартный вывод — на экран монитора, управляемого с помощью видеоконтроллера. Устройства стандартного ввода — вывода можно переназначить, в

том числе и на дисковые файлы. Стандартный вывод одной программы можно использовать как стандартный ввод для другой. При этом данные передаются через создаваемый операционной системой вспомогательный файл. Способ передачи информации от одной программы в другую называется установлением программного канала. При установке программного канала несколько программ «завязываются» в одну цепочку с автоматическим переназначением стандартных устройств ввода и вывода.

Понятие фильтра в МДОС1810 имеет особое значение. Фильтр — это программа (или команда), которая читает данные из стандартного устройства ввода, обрабатывает их и затем выводит информацию на стандартное устройство вывода. Такая программа как бы «фильтрует» данные. К числу системных фильтров относятся команды SORT, FIND и MORE.

Имеется возможность (с помощью команды COMMAND) изменить процессор команд. Новый файл с процессором команд считывается в память и становится резидентным. Это свойство широко используется при создании работающих с помощью меню прикладных систем.

В МДОС1810 широко применяются унифицированные расширения имен файлов. Так, воспринимаемые вызовы команд соответствуют файлам с расширением .EXE или .COM (в зависимости от типа загружаемого и исполняемого файла). Еще одно универсальное расширение имени файла присваивается всем командным файлам (расширение .BAT). Это приводит к тому, что в МДОС1810 отсутствует команда типа SUBMIT (имеется во многих операционных системах CM 1810, например Микрос-86). Процессор команд распознает командные файлы по расширению .BAT и переходит в пакетный режим работы. Средства организации пакетного режима в МДОС1810 довольно развиты и напоминают имеющиеся в инструментальных системах ДОС1800 и ДОС1810 (описываются в § 2.2). К ним относятся команды, исполнение которых возможно только в пакетном режиме:

- ECHO — вывод сообщения на экран консоли;
- PAUSE — пауза до нажатия клавиши на клавиатуре;
- IF — условное исполнение команд;
- FOR — повторение команд;
- GOTO — переход на другую команду в командном файле;

SHIFT — смещение формальных параметров команд;
REM — примечание.

Компоновка перемещаемых модулей, являющихся выходом трансляторов, в МДОС1810 осуществляется с помощью специальной команды LINK, которая генерирует загружаемый файл с расширением .EXE.

МДОС1810 включает в себя языки макроассемблер, БЭЙСИК, ФОРТРАН и ПАСКАЛЬ. Для операционной системы MSDOS, с которой совместима МДОС1810, имеется большое число трансляторов с других языков программирования и прикладных программ.

2.2. Инструментальные системы

Создание систем реального времени невозможно без использования мощных инструментальных средств, к числу которых, в первую очередь, относятся специальные операционные системы. Ориентированные на инструментальную область применения системы оснащаются широким набором трансляторов профессионального уровня по качеству генерируемого кода и диагностики ошибок, гибкими средствами организации пакетного режима и т. д.

Дисковая операционная система для подготовки программ реального времени на СМ 1800 (ДОС1800). В состав ДОС1800 [5] входят: супервизор, собственный монитор-отладчик, набор утилит инструментальной ориентации, экранный редактор текстов EDIT80, макроассемблер, компиляторы для языков программирования ПЛ/М и ФОРТРАН, интерпретатор БЭЙСИК, пакет программ обслуживания объектных модулей, а также универсальный макропроцессор STAGE2 и пакет программ для подготовки текстовой документации.

Макроассемблер и компиляторы вырабатывают совместимый перемещаемый объектный код, что позволяет объединять в программный комплекс различные модули, написанные на любом из входящих в состав ДОС1800 языков.

Процесс подготовки программ с помощью ДОС1800 представлен на рис. 2. 2.

Супервизор ДОС1810 выполняет следующие основные функции: обеспечивает логический доступ к файлам и устройствам; загружает программы по командам оператора и (или) по запросу программ пользователя; осуществляет построчное редактирование вводимой от оператора информации.

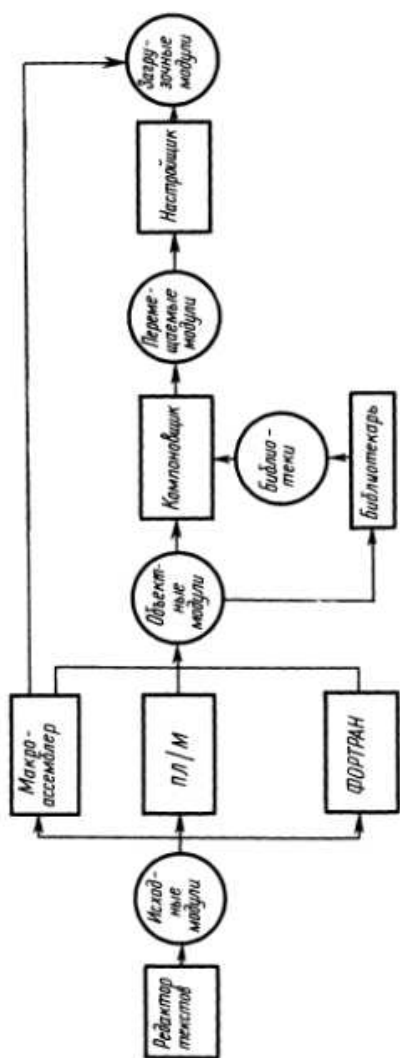


Рис. 2.2. Процесс подготовки программ с помощью DOS 1800

С точки зрения пользователя супервизор обслуживает следующие логические устройства:

:VI., :VO: — ввод — вывод для видеотерминала (дисплея);
:TI., :TO: — ввод — вывод для дополнительного терминала;
:LP: — вывод на печать;
:HR: :HP: — перфоленточный ввод — вывод;
от :F0: до :F9: — дисковые устройства.

Супервизор ДОС1800 обслуживает такое виртуальное устройство, как консоль (:CI: и :CO:), которому ставится в соответствие то или иное логическое устройство (при запуске системы — :VI: и :VO: соответственно) или файл на дискетте. Специальную функцию выполняет так называемое фиктивное устройство (:BB:), на которое выводятся ненужные данные.

Обслуживаемые ДОС1800 дискетты делятся на два типа: системные и несистемные. Системный диск содержит все специальные файлы, которые требуются для функционирования ДОС1800, с него осуществляется загрузка ДОС1800. Несистемный диск содержит больше свободного пространства.

Любая пользовательская информация, любая утилита ДОС1800 размещаются на дискеттах ДОС1800 обоих типов. Во время работы ДОС1800 на дисковом устройстве :F0: постоянно должна быть установлена системная дискетта. Физическая разметка дискетт, организация логической файловой структуры и выбор типа их выполняются в процессе работы утилиты FORMAT. Дискетта содержит до 200 файлов. Каждый файл представляет собой цепочку байтов. Место для файлов на дискетте распределяется в блоках (в ДОС1800 блок равен сектору, т. е. 128 байт). Имена файлов состоят из собственно имени (до 6 знаков) и необязательного расширения (до 3 знаков). Имя и расширения содержат алфавитно-цифровые знаки. При указании имени перед расширением ставится знак «точка». Файлам «приписываются» атрибуты (защищенный от записи, невидимый и др.). Оглавление диска содержит для каждого файла кроме указанной информации сведения о положении файлов на диске.

Супервизор ДОС1800 реализует логический доступ к файлам и устройствам при обращении пользователя к следующим системным вызовам (функциям) супервизора:

- OPEN — открытие файла;
- READ — чтение данных из файла в буфер пользователя;
- WRITE — запись данных в файл из буфера пользователя;
- SEEK — позиционирование в файле (позволяет определить текущее положение в файле или заменить текущее положение на требуемое);
- RESCAN — позиционирование в начало ранее прочитанной текстовой строки;
- CLOSE — закрытие файла;
- SPATH — получение информации о файле;
- DELETE — удаление файла из оглавления дискеты;
- RENAME — переименование файла;
- ATTRIB — изменение атрибутов дискового файла;
- CONSOL — переназначение консоли;
- WHOCON — определение текущего назначения консоли;
- ERROR — вывод сообщения об ошибке на консоль;
- LOAD — загрузка файла с исполняемыми кодами (возможно, с запуском);
- EXIT — выход из программы пользователя в супервизор DOC1800 (супервизор с помощью входящего в него интерпретатора команд ожидает от оператора имени загружаемой для исполнения или отладки программы).

Монитор DOC1800 аналогичен монитору-отладчику MONID, входящему в состав СПО. Он реализует довольно простой интерфейс с устройствами типа дисплея, печати, перфоленточного ввода — вывода, а также набора исполняемых в диалоговом режиме команд, ориентированных на процесс отладки. Набор команд монитора DOC1800 включает в себя следующие группы команд:

- управления памятью (D — вывод содержимого памяти, F — заполнение области памяти константой, M — пересылка области памяти, S — изменение содержимого ячейки памяти), вся эта группа команд эквивалентна соответствующим одноименным командам программы MONID:

- управления регистрами микропроцессора (X), эквивалентна команде X монитора-отладчика MONID.

- команды управления конфигурацией ввода — вывода (A — назначение конфигурации ввода — вывода, Q — отображение конфигурации);

- ввода — вывода перфоленты;

- выполнения (G — запуск программы), отличается от команды G в MONID тем, что имеется возможность останова не по одной, а по двум точкам.

Мониторные функции обеспечивают интерфейс с системными периферийными устройствами, в том числе опрос состояния. К ним можно отнести следующие функции:

CI — ввод с консоли;

CO — вывод на консоль;

CSYS — получение состояния клавиатуры консоли (определяет, была ли нажата клавиша);

RI — ввод из устройства ввода;

RO — вывод на устройство вывода;

LO — вывод на устройство вывода листинга.

Имеется также ряд вспомогательных функций.

При выполнении любой программы (команды) DOS1800 вводится ее имя. Если программу требуется отладить, то перед именем следует указать слово DEBUG, означающее необходимость (после загрузки программы) передачи управления в монитор DOS1800.

Точно так же выполняются и утилиты DOS1800, которые условно можно разделить на следующие группы:

— обслуживания дискетт;

— обслуживания файлов;

— обслуживания текстовых файлов;

— организации пакетного режима работы;

— прочие утилиты.

К группе утилит обслуживания дискетт относятся:

DCOPY — копирование;

PATCH — коррекция;

FORMAT — инициализация (форматирование);

RELAB — переименование;

DIR — вывод оглавления (каталога);

DSORT — сортировка оглавления.

К группе утилит обслуживания файлов относятся:

COPY — копирование;

RENAME — переименование;

DELETE — удаление;

ATTRIB — изменение атрибутов.

К группе утилит обслуживания текстовых файлов относятся:

PACK — уплотнение;

UNPACK — разуплотнение;

COMPAR — сравнение;

SORT — сортировка.

В ДОС1800 имеется возможность пере назначить консоль (:CI: и :CO:), что позволит реализовать пакетный режим работы, при котором вместо консоли входная информация берется из файла (так называемого командного файла) на ГМД (дискете). Организация пакетного режима осуществляется с помощью утилит SUBMIT, EXEC и CONSOL:

SUBMIT позволяет помимо пере назначения консоли осуществить присвоение фактических значений формальным параметрам, которые в командном файле задаются как знак % с цифрой (номером параметра);

EXEC отличается от SUBMIT тем, что в командном файле задается один формальный параметр (%0), вместо которого последовательно подставляются все фактические значения, указанные при вызове утилиты EXEC. Таким образом, последовательность команд в командном файле выполняется столько раз, сколько указано фактических значений параметров (это удобно, например, при выполнении одинаковых действий типа трансляции для набора файлов);

CONSOL пере назначает :CI: и :CO:, подстановка значений параметров отсутствует.

Пакетный режим обладает большой гибкостью, обеспечиваемой утилитами, которые используются только внутри командных файлов:

IF — проверка условия (существование файла, равенство файлов и т. д.);

ELSE — иначе, для утилиты IF;

ENDIF — конец блока IF в командном файле;

GOTO — переход на метку в командном файле (метка — это строка командного файла, начинающаяся со знаков ;; и имеющая тот же формат, что и имя файла);

RETURN — возврат из командного файла;

EXIT — выход из пакетного режима;

LOOP — цикл (с начала командного файла);

RESCAN — перезапуск командного файла;

WAIT — ожидание вмешательства оператора;

NOTE — комментарий в командном файле;

STOPIF — окончание пакетного режима при обнаружении ошибок во время трансляции.

К числу прочих утилит относятся:

VERS — вывод номера версии программы;

CLEAN — удаление старых версий;

LATEST — вывод номера последних версий.

В состав ДОС1800 входит экранный редактор текстов EDIT80. Термин «экранный» означает, что редактирование текстовой информации осуществляется на экране дисплея с помощью клавиш управления курсором. Часть текстового файла отображается на экране. С помощью клавиш ←, →, ↑, ↓ оператор устанавливает курсор на требуемое место в текстовом файле. Далее выполняются действия по редактированию. К ним относятся (в скобках приводятся соответствующие клавиши):

- замена текста (новый текст вводится без нажатия каких-либо специальных клавиш редактирования, по мере ввода текст знак за знаком заменяет старый текст, отображенный на экране);

- вставка знака (УС + С);

- вставка текста (УС + А, содержимое экрана от местоположения курсора до конца «чистится», оператор вводит новый текст, в конце которого требуется снова нажать УС + А, при этом введенный текст оказывается вставленным в то место, где было введено в первый раз УС + А, старый текст смещается за новый текст, т. е. после вставки);

- уничтожение знака (УС + D);

- уничтожение текста (УС + Z, знак, указываемый курсором, заменяется на @, что отмечает начало удаляемого текста, с помощью клавиш управления курсора второй указатель @ перемещается в то место, где заканчивается удаляемый текст, при этом оператор должен снова нажать клавиши УС + Z, происходит удаление текста).

Для перемещения по текстовому файлу имеются управляющие клавиши УС + N (отображение на экране следующей части текстового файла), УС + P (отображение на экране предыдущей части текстового файла), УС + V (отображение на экране текущей части текстового файла).

Редактор текстов EDIT80 помимо экранного режима имеет также командный режим. Основными командами для редактирования текстовой информации являются следующие:

- J — переход по текстовому файлу;

- DL и DC — исключение строк и знаков;

- I — вставка текста;

- XМ и XС — перемещение и копирование части текста;

- F и S — поиск и замена знаков по контексту;

R и W — чтение и запись текста для дискового файла;

G — выбор команд для редактора текстов из дискового файла;

TS — установка тэгов, отмечающих то или иное место в текстовом файле;

EX — выход из редакторов текстов в ДОС1800.

Очень просто можно осуществлять переход из экранного режима в командный (путем нажатия клавиши HOME или \curvearrowright) и наоборот (путем нажатия клавиши $UC + V$).

Дополнительные средства редактирования текстов включают в себя макросредства и средства настройки на дисплей.

Макросредства позволяют в командном режиме определить последовательность действий командного и (или) экранного режима как один макрос с именем. Далее эта последовательность команд может быть вызвана как единое целое. Если макрос предназначен для работы в экранном режиме и имеет имя, соответствующее нажатию клавиши UC вместе с какой-либо другой клавишей, то вызов этого макроса осуществляется нажатием клавиши его имени. Таким образом, можно создать макросы, расширяющие возможности EDIT80 (например, $UC + S$ — перемещение в начало строки, $UC + E$ — перемещение в конец строки, $UC + T$ — перемещение в начало текстового файла и т. д.). В целом макросредства позволяют создавать на базе EDIT80 проблемно-ориентированный язык редактирования текстов, например для ввода и коррекции исходных текстов программ на языке ПЛ/М, создания текстовой документации и т. д.

Дисплеи различных типов имеют разные значения управляющих кодов, в том числе и для перемещения курсора. С помощью специального набора команд редактор текстов EDIT80 перенастраивается путем изменения (переопределения) кодов и функций: кодов управляющих клавиш, имен команд экранного режима, кодовых последовательностей для чистки экрана и т. д.

В ДОС1800 принят единый формат объектных модулей, которому соответствуют как выходные модули, генерируемые различными трансляторами, так и библиотечные и загрузочные модули. Структура и организация объектного модуля в ДОС1800 не зависит от языка, на котором написан соответствующий исходный модуль.

Программа LINK предназначена для объединения нескольких объектных модулей в один. Наличие LINK

в составе ДОС1800 позволяет реализовать концепцию модульного программирования. Связь между модулями возможна на уровне как подпрограмм (процедур), так и общих данных. Для определения таких связей объектам исходного текста присваиваются атрибуты: PUBLIC (глобальный объект) и EXTERNAL (внешний объект). Ссылки на внешние объекты (внешние ссылки) разрешаются на этапе компоновки с помощью программы LINK. Входной информацией для нее служат как объектные модули, так и библиотеки объектных модулей (см. далее), что способствует применению современных технологий программирования.

Программа LOCATE настраивает объектный модуль на конкретный адрес памяти, формируя тем самым готовый к исполнению загрузочный модуль.

Библиотекарь LIB позволяет создавать библиотеки объектных модулей. При компоновке в формируемый выходной объектный модуль из указанных библиотек подключаются только те библиотечные объектные модули, на которые имеются внешние ссылки.

Дисковая операционная система для подготовки программ реального времени на СМ 1810 (ДОС1810). ДОС1810 полностью совместима с ДОС1800. Для ее работы в СМ 1810 помимо 16-разрядного процессора МЦП-16 необходимо устанавливать 8-разрядный процессор МЦП-1. При этом ДОС1810 работает как в 8-разрядном, так и в 16-разрядном режимах. В 8-разрядном режиме вся работа ДОС1810 осуществляется на МЦП-1. В 16-разрядном режиме основным процессором является МЦП-16, а МЦП-1 выполняет функции процессора ввода — вывода.

Основными отличиями ДОС1810 от ДОС1800 являются:

- более широкая номенклатура обслуживаемых устройств;

- расширенные возможности обработки командных строк в супервизоре;

- наличие 16-разрядного режима работы.

Рассмотрим эти отличия более подробно.

В ДОС1810 обслуживаются следующие новые дисковые контроллеры:

- СМ 1810.5125 — контроллер ГМД;

- СМ 1810.5124 — контроллер дисков СМ5408;

- СМ 1810.5126 — контроллер дисков типа «Винчестер» и ГМД;

СМ 1810.5123 — контроллер жестких дисков с интерфейсом СМД.

Обслуживается также устройство внешней памяти на ГМД с прямым доступом СМ 5635.10 из состава СМ 1803.09.

Основной тип ГМД — гибкие диски диаметром 133 мм, например СМ 5640 (односторонние, двойной плотности, 80 дорожек).

Большинство типов дисков, используемых в ДОС1810, имеют физическую организацию с длиной сектора, большей 128 байт (как было принято в ДОС1800), поэтому понятия физического и логического секторов требуют разделения. Логические секторы (блоки) имеют длину 128 байт, необходимую для совместимости с ДОС1800. Физические секторы содержат в зависимости от типа накопителя 1—4 блока. Это приводит к необходимости буферизации обмена с дисками. Если какой-либо блок, запрашиваемый для чтения супервизором ДОС1810, уже содержится в некотором буфере, то этот блок возвращается супервизору без обращения к диску. Такой обмен потребовал введения алгоритма распределения буферов, определяющего, какой из них должен освобождаться первым при нехватке буферов.

Периферийные устройства (консоль и печатающее устройство) подключаются к СМ 1810 через последовательные каналы на модулях МИРПС-М, МСМ или МСТ. Использование интерфейсов ИРПС (в модулях МИРПС-М и МСТ) и СТЫК С2 (в модуле МСМ) упрощает физический обмен и программную реализацию обмена.

В ДОС1810 предусмотрена возможность гибкого переопределения обслуживаемых устройств с помощью команды ASSIGN, позволяющей поставить в соответствие любому дисковому устройству (от :F0: до :F9:) тот или иной накопитель, подключенный через соответствующий контроллер. Возможно также переопределение дисплея (обслуживается через МИРПС-М или МСМ) и печатающего устройства (обслуживается также через МИРПС-М или МСМ). С точки зрения программиста модули МСМ и МСТ идентичны. При определении устройства в команде ASSIGN можно указать все свойственные ему физические характеристики для правильного и эффективного обслуживания супервизором ДОС1810.

Дополнительным устройством, обслуживаемым супервизором ДОС1810, является «электронный» диск, представляющий собой назначаемую оператором область опе-

ративной памяти СМ 1810, которая логически организована как диск, хранящий файлы.

По аналогии с операционными системами общего назначения вводятся встроенные команды. Появились также средства для вызова и редактирования предыдущей (введенной) команды. К числу важнейших встроенных команд относятся следующие:

- Ⓐ — отображение текстового файла на экран консоли;
- — исполнение однострочного командного файла;
- / — исполнение простого (т. е. без формальных параметров) командного файла;

JOB — создание и исполнение простого командного файла.

Вызов 16-разрядного режима работы ДОС1810 осуществляется с помощью команды **RUN**. В этом режиме СМ1810 начинает работать как духпроцессорная микро-ЭВМ. Основным процессором является 16-разрядный МЦП-16; 8-разрядный процессор МЦП-1 выполняет функцию процессора ввода — вывода.

В зависимости от способа вызова команды **RUN** исполнение в 16-разрядном режиме может быть интерактивным (т. е. оператор вызывает программу **RUN** и далее выполняет обращение к программам для 16-разрядного процессора) или неинтерактивным (в команде **RUU** непосредственно указывается имя одной программы, которую необходимо исполнить в 16-разрядном режиме). Имеются следующие подкоманды для **RUN**:

DATE — установ даты;

WORK — определение устройств, используемых для хранения рабочих файлов;

EXIT — выход в 8-разрядный режим работы ДОС1810, т. е. в интерпретатор команд как часть супервизора.

В 16-разрядном режиме могут исполняться:

— макроассемблер;

— трансляторы с языков ПЛ/М-86, ФОРТРАН-86 и ПАСКАЛЬ-86;

— компоновка (**LINK 86**), настройка на адрес (**LOC86**) и ведение библиотек объектных модулей (**LIB86**);

— программа составления таблиц межмодульных перекрестных ссылок (**CREF86**).

В ДОС1810 имеется библиотека обслуживания арифметического сопроцессора.

Для выполнения системных функций программа в 16-разрядном режиме должна обращаться к вызовам, входящим в состав универсального программного интерфейса (см. Приложение).

Для отладки программ, работающих на МЦП-16, используется отладчик DEBUG-86, который позволяет:

- осуществлять контроль и изменение регистров и флагов микропроцессора K1810VM86;
 - выводить содержимое памяти, в том числе в ассемблерной мнемонике;
 - запускать программу с введением точек останова;
 - использовать символические имена в процессе отладки (т. е. вести отладку в терминах исходного текста программ);
 - организовывать составные команды, выполнять проверки условий, ветвление, итеративные циклы и т. д.
- Физически отладчик DEBUG-86 размещен в постоянной памяти МЦП-16.

2.3. Операционные системы реального времени

МикроЭВМ CM1800 и CM1810 широко используются в системах реального времени, предназначенных как для АСУ ТП, так и для лабораторных применений. Программное обеспечение этих применений основывается на ряде операционных систем реального времени, работающих в мультипрограммном режиме и оснащенных набором драйверов устройств, входящих в состав микроЭВМ.

Общие сведения об операционных системах реального времени для CM 1800. Программное обеспечение микроЭВМ CM 1800 включает в себя три ОС реального времени [5]: базовую резидентную — БРС РВ; мультипрограммную — МОС РВ; мультимикропроцессорную со специализацией функций процессоров — ОС СФП.

Эти ОС совместимы, основываются на одних и тех же архитектурных принципах и являются чисто исполнительными, т. е. не включают в себя инструментальной поддержки. Разработка программ, работающих под управлением БРС РВ, выполняется с помощью СПО; разработка программ, работающих под управлением МОС РВ и ОС СФП, — с помощью ДОС1800.

По своим характеристикам рассматриваемые ОС являются мультипрограммными многозадачными с диспетчеризацией на основе приоритетов. В состав их входят ядро и драйверы. Организация взаимодействия задач

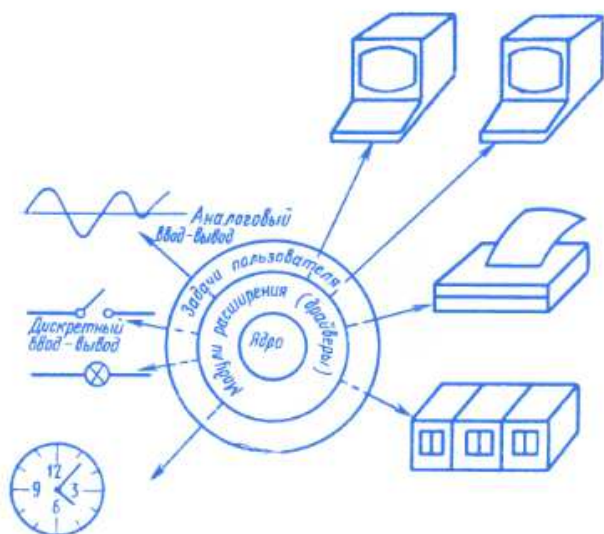


Рис. 2.3. Обобщенная структура ОС реального времени для CM 1800

с ядром такова, что все компоненты систем, в том числе драйверы, оформляются как задачи, работающие под управлением ядра ОС реального времени. Обобщенная структура ОС реального времени для CM 1800 представлена на рис. 2.3.

Хотя указанные ОС совместимы, между ними имеются некоторые отличия, для иллюстрации которых перечислим основные характеристики ОС реального времени.

БРС РВ включает в себя ядро, простейшие драйверы периферийных устройств (терминала и печати), драйверы устройств связи с объектом (дискретного и аналогового ввода — вывода, ввода число-импульсных сигналов), а также строчно-ориентированный эмулятор терминала как средство межмашинного обмена информацией. Инструментальная поддержка БРС РВ находится в операционной среде СПО. БРС РВ поставляется как законченная и настроенная на абсолютный адрес система на дискетках формата СПО.

МОС РВ по сравнению с БРС РВ имеет расширенные функциональные возможности драйверов (при сохранении совместимых версий). В состав МОС РВ включены такие новые компоненты, как:

- распределитель памяти (для выделения и утилизации буферов из общего пространства свободной памяти);
- отладчик (объект отладки — задача, т. е. при отладке действия выполняются для конкретной задачи параллельно с выполнением остальных задач в реальном масштабе времени);
- файловая система (обеспечивает ведение файлов на дискеттах формата ДОС1800 в реальном времени);
- начальный загрузчик (создает минимальные по объему версии систем реального времени для размещения в постоянную память, которые, в свою очередь, загружают полноценную систему реального времени из файла на дискетте);
- драйвер управления информационным каналом (эффективное помехозащищенное программное средство обмена данными с другой СМ 1800 или с ЭВМ типа СМ 4, СМ 1420).

Следует отметить, что МОС РВ поставляется как набор библиотек на дискеттах формата ДОС1800. С помощью ДОС1800 как инструментальной системы пользователь создает собственные прикладные задачи. Объединение пользовательских объектных модулей с библиотечными модулями компонентов МОС РВ осуществляется путем компоновки программой LINK из состава ДОС1800. Таким образом, МОС РВ — генерируемая система, в которую пользователь включает только модули, необходимые для функционирования собственной системы реального времени.

ОС СФП являются дальнейшим развитием МОС РВ. Основное отличие ОС СФП — возможность реализации мультимикропроцессорного режима. Специфика ОС СФП заключается в том, что на каждом из процессоров (модуле МЦП-1 из состава микроЭВМ СМ 1804) функционирует собственный вариант системы реального времени, включающий ядро, необходимые драйверы и пользовательские задачи. Взаимодействие между процессорами реализуется специальным компонентом ОС СФП, который обеспечивает обмен данными (информацией) между задачами, выполняемыми различными процессорами. Таким образом, на этапе создания системы пользователь определяет, какие из задач решаются на том или ином процессоре, осуществляя статическую специализацию функций, выполняемых процессорами.

В ОС СФП функциональные возможности некоторых компонентов расширены. Ядро ОС СФП работает как

на процессоре МЦП (из состава СМ1803, в однопроцессорном режиме), так и на процессоре МЦП-1 (из состава СМ 1804, в однопроцессорном и мультимикропроцессорном режимах). Имеются новые драйверы (например, драйвер модуля вывода символьной информации на телемонитор).

ОС СФП полностью совместимы с МОС РВ, что облегчает перенос прикладных систем из среды МОС РВ в мультимикропроцессорную операционную среду ОС СФП.

Для микроЭВМ СМ 1810 разработана операционная система ОС СФП1810, функционально аналогичная ОС СФП для СМ 1800.

Большая операционная система (БОС1810). Для СМ 1810 кроме чисто исполнительно-операционной системы реального времени ОС СФП1810 имеется эффективная мультипрограммная операционная система реального времени БОС1810, сочетающая в себе как исполнительные, так и инструментальные возможности. Фактически БОС1810 одновременно является системой реального времени и системой разделения времени.

Структура БОС1810 приведена на рис. 2.4.

Рассмотрим архитектурные особенности и возможности БОС1810.

Объектно-ориентированная архитектура. БОС1810, как и любая операционная система, является набором программных средств, используемых программистами. Подобно языкам программирования, строгая классификация объектов в БОС1810 облегчает понимание системы, превращает ее в набор строительных блоков для конструирования прикладных систем. Каждый объект имеет определенные атрибуты, общие для всех объектов данного типа, поэтому для изучения важно знание их. Пользователь при освоении БОС1810 освобожден от необходимости знания подробностей об объектах, которые он не планирует применять.

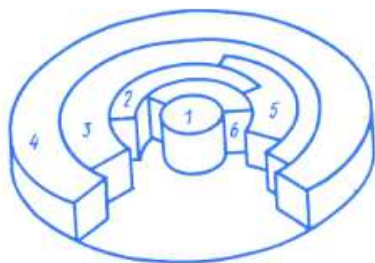


Рис. 2.4. Структура БОС 1810:

1 — ядро; 2 — расширенная система ввода — вывода; 3 — интерфейс с оператором; 4 — универсальный программный интерфейс; 5 — загрузчик прикладных программ; 6 — базовая система ввода — вывода

Многозадачность. Для упрощения разработки систем реального времени в БОС1810 реализована концепция многозадачности. Действительно, сущность обработки в реальном масштабе времени состоит в обработке многочисленных внешних асинхронных событий. Любая однопрограммная система, взаимодействующая с асинхронными событиями, имеет степень сложности, пропорциональную числу обрабатываемых типов событий. Поэтому сложные системы реального времени обычно реализуются как многозадачные. Вместо одной сложной программы целесообразно разрабатывать множество программ для обработки небольших групп событий или даже одиночных событий.

Обработка прерываний. Существует два способа реакции на внешние события: опрос и обработка прерываний. Опрос — чисто программная периодическая проверка наличия фактов внешних событий; обработка прерываний — реакция процессора на внешнее событие с прерыванием текущей работы. Второй способ обладает большими производительностью и гибкостью.

Диспетчеризация на основе приоритетов. БОС1810 использует метод диспетчеризации для определения задачи (программы), которая должна исполняться в текущий момент времени. Суть этого метода заключается в том, что если более важная задача становится готовой к исполнению (например, из-за того, что произошло ожидаемое ею внешнее событие), то менее важная задача прерывается, а более важная начинает немедленно выполняться. Заметим, что используемый в системах разделения времени метод квантования времени малопригоден для построения эффективных систем реального времени.

Мультипрограммирование. Это понятие близко к многозадачности. Однако цель мультипрограммирования иная — более полное использование вычислительных ресурсов. Мультипрограммирование означает, что в системе реализуется несколько применений. Для достижения эффективности необходимо обеспечить разделение операционной среды для каждого применения (разделение памяти, файлов и объектов), с тем чтобы изолировать их. БОС1810 содержит объект, позволяющий легко изолировать применения друг от друга. Этот объект (задание) включает в себя набор задач и требуемых для их исполнения ресурсов. Задания обычно соответствуют отдельным применениям.

Межзадачная координация. Решение задач координации для реализации взаимоисключения, синхронизации и взаимодействия являются важными для всех операционных систем реального времени CM 1800 и CM 1810. БОС1810 предоставляет простые методы для координации задач. Обычно обмен информацией удобно реализовывать с помощью почтовых ящиков, иногда называемых также обменниками. Для обеспечения взаимоисключений широко применяются объекты типа семафоров и регионов. Синхронизация как проблема межзадачной координации лучше всего решается с помощью семафоров.

Возможность расширения. БОС1810 — расширяемая система. Принцип расширяемости распространяется и на объекты. Пользователь может создавать собственные типы объектов, при этом добавляя новые системные вызовы к БОС1810.

Возможности отладки. К числу компонентов БОС1810, реализующих процесс отладки, относятся динамический и системный отладчики. Оба отладчика ориентированы на объекты БОС1810 (задачи, почтовые ящики и т. д.), что существенно упрощает создание сложных систем реального времени.

Системы ввода — вывода. В БОС1810 имеется две системы ввода — вывода: базовая и расширенная.

Базовая система ввода — вывода повышает производительность конечной прикладной системы, однако за счет увеличения времени на ее проектирование. Мощность базовой системы ввода — вывода такова, что позволяет:

- создавать пользовательский алгоритм буферизации;
- организовывать прямой доступ к данным в файлах;
- гибко описывать детали системных вызовов, что дает возможность настраивать базовую систему ввода — вывода на требования прикладной системы.

Расширенная система ввода — вывода эффективна при последовательном доступе к файлам, она выполняет функции автоматизации буферизации.

Независимый ввод — вывод. Системные вызовы БОС1810 не зависят от устройств, а сами устройства фигурируют как параметры системного вызова.

Иерархическая файловая структура. Применение иерархических файловых структур получает

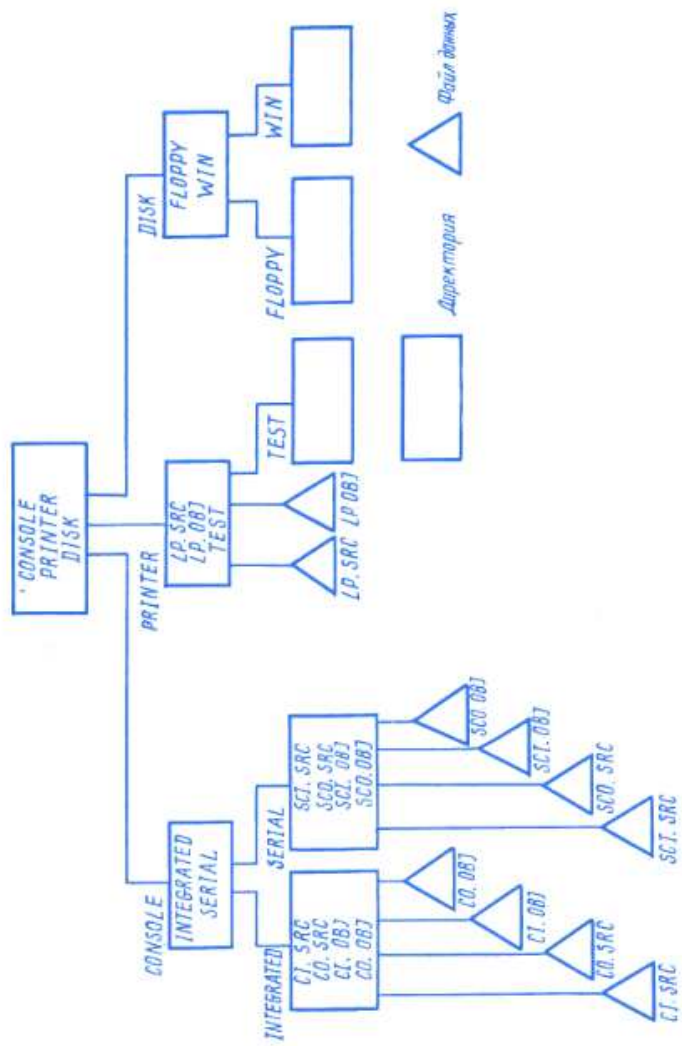


Рис. 2.5. Пример иерархического наименования файлов

все более широкое распространение в современных операционных системах. На рис. 2.5 приведен пример иерархического наименования файлов.

Иерархическая файловая структура упрощает добавление новых применений, делает систему более удобной для работы нескольких независимых операторов. При доступе к иерархически именованным файлам применяется контроль доступа (т. е. прав доступа). Это позволяет, в частности, более строго отделять одно применение от другого. По сравнению с более простыми или одноуровневыми каталогами принципы структуризации исходно закладываются в организацию файлов как наборов данных.

Управление фрагментацией файлов. При разработке прикладных систем важно найти оптимальное значение производительности и требуемого объема памяти. В БОС1810 пользователь имеет возможность определять фрагментацию файлов, т. е. размер блоков, из которых состоят файлы. Фрагментация влияет на скорость передачи информации, время доступа к ней, а также на размер используемого в системных целях дискового пространства.

Многопользовательский интерфейс с оператором. БОС1810 — многопользовательская операционная система. Эта возможность реализуется с помощью специальной компоненты системы — интерфейса с оператором, работающего в многотерминальном режиме. БОС1810 позволяет оператору с терминала (дисплея) выполнять команды вызова утилит, компиляторов и т. д. Операционная система определяет момент включения терминала и назначает рабочую среду для его обслуживания, включающую идентификатор, область памяти для исполнения программ и приоритет. Стандартно после этого запускается интерпретатор командных строк, который осуществляет грамматический разбор введенных оператором команд, выделяет из них имя запускаемой программы, загружает ее и напускает, передавая программе «остаток» командной строки, содержащей параметры. При этом пользователь имеет возможность определить собственную программу, запускаемую интерфейсом с оператором вместо интерпретатора строк команд.

Обработка ошибок. Обработка ошибок как процесс выявления и реакции на неожиданные ситуации в значительной степени выполняется компонентами

БОО1810. Операционная система защищает прикладное программное обеспечение от большинства типов ошибок. При этом в системе используются понятия кода возврата и драйвера особых ситуаций. Под драйвером особых ситуаций понимается процедура, которую можно вызвать, когда задача после исполнения системного вызова (например, ядра БОО1810) получает код возврата, идентифицирующий ошибку при исполнении системного вызова. Для каждой задачи имеется возможность определять собственный драйвер особых ситуаций, написанный пользователем или являющийся стандартным из состава БОО1810.

Динамическое распределение памяти. Такое распределение памяти (в процессе работы системы) позволяет уменьшать общий объем требуемой оперативной памяти; разделять память между заданиями; разделять память между задачами внутри задания.

Оперативная память в БОО1810 выделяется по мере поступления запросов на ее распределение. После того как требуемое действие произошло и память больше не нужна, следует вернуть ее, сделав доступной для дальнейшего распределения.

Привязка во время исполнения. Она дает возможность разделять объекты в различных заданиях; использовать в процедурах логические имена файлов и устройств; упрощать процесс объединения прикладного программного обеспечения с компонентами БОО1810.

При этом под привязкой понимается процесс указания каждой программе расположения используемых ею процедур и переменных. Частично привязка выполняется на этапе компиляции программы, а также в процессе компоновки. Следующий ряд привязок выполняется только во время исполнения:

- привязка объектов к задачам;
- привязка файлов и устройств к задачам;
- привязка прикладных программ к ОС.

Рассмотрим привязку во время исполнения на следующем примере. Пусть две задачи используют для обмена информацией между собой какой-то почтовый ящик. Но если программы для двух задач скомпилированы (оттранслированы) отдельно и скомпонованы независимо одна от другой (что имеет место, если они находятся в разных заданиях), то задачи используют привязку для организации обращения к одному и тому же почтовому ящику.

Привязка объектов во время исполнения состоит в том, что, когда задача создает разделяемый с другими задачами объект, он должен быть занесен в каталог объектов. Другие задачи могут использовать этот каталогизированный объект, если им известно имя объекта как цепочка символов.

Интерактивное конфигурирование. БОС1810 — конфигурируемая система. Пользователь может выбрать только те части (компоненты) БОС1810,

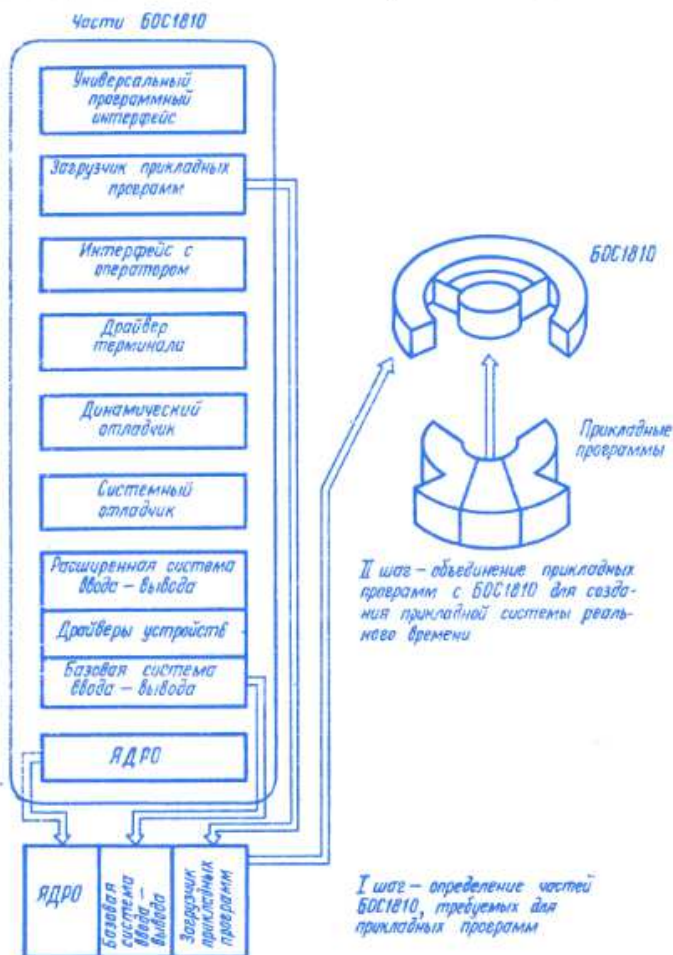


Рис. 2.6. Интерактивное конфигурирование БОС 1810

которые ему требуются. Конфигурирование выполняется с помощью специального компонента — интерактивного конфигуратора. Конфигурирование — это последовательный процесс: сначала выбираются используемые компоненты БОС1810 и их характеристики, затем модули БОС1810 объединяются с прикладными программами (рис. 2.6). Этот процесс является диалоговым (интерактивным).

Рассмотрим функции компонентов (слоев) БОС1810, включаемых в прикладную систему в процессе конфигурирования.

1. Ядро — основной компонент БОС1810, входящий в любую прикладную систему, выполненную на основе БОС1810. Некоторые характеристики ядра более подробно рассмотрены далее.

2. Базовая и расширенная система ввода — вывода — не обязательные системы. Они обеспечивают управление файлами и независимый от устройств интерфейс ввода — вывода. Пользователь может включать базовую систему ввода — вывода и не включать расширенную; включение расширенной системы ввода — вывода всегда требует включения базовой системы ввода — вывода.

3. Драйверы устройств — интерфейс с устройствами ввода — вывода. Любые выбранные при конфигурировании драйверы (включая терминальные) становятся частью базовой системы ввода — вывода.

4. Интерфейс с оператором позволяет управлять прикладной системой посредством команд оператора, вводимых с терминала. Включает в себя набор команд для наиболее широко используемых операций. Имеется возможность создания пользователем собственных команд. Интерфейс с оператором может в прикладной системе отсутствовать, но если он присутствует, то требуются все остальные слои.

5. Загрузчик прикладных программ позволяет загружать программы и их оверлеи с диска. При включении загрузчика требуется наличие систем ввода — вывода.

6. Динамический отладчик — не обязательный, но полезный компонент, служащий для отладки прикладной системы. Окончив отладку, можно удалить его, переконфигурировать систему.

7. Системный отладчик также не обязательный компонент, расширяющий возможности размещенного в постоянную память системного монитора-отладчика (МОНИТОР-86). Обеспечивает «статическую» отладочную

информацию о системе после отказа или в любой момент, когда надо «заморозить» и проверить систему.

8. Драйвер терминала. Позволяет обращаться к терминалу без применения систем ввода — вывода или интерфейса с оператором.

9. Универсальный программный интерфейс (УПИ) — интерфейс, позволяющий инструментальным программным средствам (включая языковые трансляторы) обращаться к средствам БОС1810. УПИ — самый верхний слой любой прикладной системы, может и не включаться, если в этом нет необходимости. При включении требует наличия интерфейса с оператором и всех других слоев БОС1810. Краткие сведения об УПИ приводятся в Приложении.

Команды оператора. При разработке прикладной системы часто возникает необходимость в работе с файлами. Помимо возможности написания собственных программ для этих целей БОС1810 предоставляет готовые для выполнения программы. Эти команды выполняются как команды оператора.

Команды оператора делятся на следующие группы: управления файлами; управления томами; многопользовательские; вспомогательные.

К основным командам управления файлами относятся:

- ATTACHFILE — связывание логического имени с существующим файлом;
- COPY — создание или копирование файлов;
- CREATEDIR — создание директорий;
- DELETE — удаление файлов данных или пустых директорий из тома;
- DETACHFILE — прекращение связи логического имени с файлом;
- DIR — вывод списка файлов (с атрибутами), содержащихся в директории;
- PERMIT — разрешение — запрет доступа к файлам;
- RENAME — переименование файлов и директорий.

В число основных команд управления томами входят:

- ATTACHDEVICE — присоединение физического устройства к системе и связывание этого физического устройства с логическим именем;
- BACKUP — копирование файлов в резервируемый том;
- DETACHDEVICE — удаление физического устройства из системы;
- DISKVERIFY — проверка структуры тома;

- FORMAT — форматирование тома;
- RESTORE — копирование файлов из резервируемого тома.

К многопользовательским командам относятся следующие:

- INITSTATUS — вывод статуса терминалов;
- JOBDELETE — удаление текущего интерактивного задания;
- LOCK — запрет создания интерактивного задания;
- SUPER — изменение идентификатора пользователя.

Вспомогательные команды содержат следующие основные команды:

- DATE — установ или изменение даты;
- DEBUG — вызов монитора-отладчика МОНИТОР-86;
- LOGICALNAMES — вывод всех логических имен в системе;
- MEMORY — вывод доступного пользователю объема памяти;
- PATH — вывод полного имени файла;
- SUBMIT — организация пакетного режима работы;
- TIME — установ или изменение времени;
- VERSION — вывод номера версии команды.

Инструментальная поддержка. В составе БОС1810 имеются развитые языковые трансляторы, такие, как макроассемблер, ПЛ/М-86, ФОРТРАН-86 и ПАСКАЛЬ-86. Получаемые в результате работы перемещаемые объектные модули могут быть объединены (с помощью компоновщика LINK86), настроены на адрес (с помощью настройщика LOC86) или включены в библиотеки (с помощью библиотекаря LIB86). Возможно также составление таблицы межмодульных перекрестных ссылок (программа CREF86). Имеется также библиотека поддержки арифметического сопроцессора.

Все перечисленные средства инструментальной поддержки полностью совместимы со средствами, работающими в 16-разрядном режиме в ДОС1810.

Дополнительным свойством этих средств является то, что в БОС1810 они могут работать в многопользовательском режиме, т. е., например, три пользователя могут со своих терминалов одновременно транслировать программы с помощью макроассемблера.

В состав БОС1810 входят также программы для выпуска текстовой документации, являющиеся расширением пакета для ДОС1800.

Экранный редактор текстов. В БОС1810 более развиты средства для занесения и редактирования

текстовых файлов, в том числе исходных текстов программ. Для этих целей в БОС1810 используется управляемый с помощью подсказок экранный редактор текстов EXTED (существуют также версии этой программы, работающие под управлением ДОС1810 как в 16-разрядном, так и в 8-разрядном режимах).

При работе редактора EXTED нижняя строка экрана дисплея используется для подсказок. Имеется три вида подсказок: меню; подсказки, требующие построчно-редактируемого ввода подсказки; подсказки, требующие ответа ДА—НЕТ (YES/NO подсказки).

После вызова EXTED выводится подсказка — меню и редактор текстов находится на главном командном уровне. Меню — это часть общего списка ключевых слов, обозначающих имена возможных команд. Одновременно может отображаться до восьми имен команд. Для выбора требуемой команды необходимо нажать клавишу, соответствующую первой букве в имени команды. Возможен ввод и тех команд, которые в данный момент не входят в отображаемую часть общего списка команд. Для просмотра списка имен допустимых команд (ключевых слов) надо нажать клавишу TAB (или GT) на клавиатуре дисплея.

Для перемещения по тексту (он отображается в верхних 22 строках экрана) используются клавиши управления курсором ←, →, ↑ и ↓. Если курсор перемещается за пределы отображаемого на экране текста, то происходит «сдвиг» изображения таким образом, чтобы на экране всегда отображалась часть текстового файла, содержащая знак, указываемый курсором.

Установив курсор на нужный знак в тексте, оператор может выполнить одну из следующих команд редактора EXTED:

- INSERT — режим вставки (завершается нажатием клавиши ESC или AP2);
- XCHANGE — режим замены (завершается нажатием клавиши ESC или AP2);
- FIND — поиск (выдается подсказка — запрос искомого текста);
- FIND — поиск в обратном направлении (т. е. назад, к началу файла);
- REPLACE — поиск и замена (выдается подсказка — запрос заменяемого и заменяющего текстов);
- ? REPLACE — поиск и замена, но с выдачей требующей ответа под-

- сказки, оператор должен ввести подтверждение необходимости замены;
- TAG** — установ тэгов (т. е. меток в тексте) от A до D. Выбор имени тэга — с помощью специального меню;
- JUMP** — переход по тексту. Возможен переход в начало (START) текста, конец (END) текста, на требуемую строку (LINE) или в позицию на строке (POSITION), а также на тэги (от A до D). Выбор типа перехода — с помощью специального меню;
- BLOCK** — определение блока (участка) текста. После отметки начала и конца блока его можно поместить в специальный файл (PUT) или буфер (BUFFER), удалить (DELETE). В процессе отметки блока можно искать (FIND или —FIND) текст, осуществлять переход (JUMP). Все указанные подкоманды задаются в общем специальном меню;
- DELETE** — определение удаляемого блока;
- GET** — чтение содержимого файла (или буфера блока) и помещение его в место, указываемое курсором;
- VIEW** — перезапись содержимого экрана, при этом строка с курсором (в текстовом файле остается в прежнем месте) чаще всего оказывается на экране пятой;
- OTHER** — переход к редактированию другого (альтернативного) текстового файла. На экране начинается процесс редактирования другого (из двух обрабатываемых) файлов;
- AGAIN** — повторение предыдущей команды;
- SET** — выполнение действий по выбору режима работы EXTED. Возможна установка табуляторов, левой границы текста и других опций;
- HEX** — шестнадцатеричный ввод или вывод;
- QUIT** — выход из EXTED. Выдается специальное меню, позволяющее прервать (ABORT) сеанс редактирования без выполнения записи текста в файл, выйти (EXIT) из редактора с записью в файл скорректированного текста, начать (INIT) новый сеанс по редактированию другого файла, изменить (UPDATE) содержимое файла без прерывания сеанса редактирования, записать (WRITE) текст в указанной оператором файл.

В редакторе текстов EXTED перед именем команды можно указывать повторитель (определяет число исполнений последующей команды); можно определять собственные проблемно-ориентированные средства редактирования, создаваемые с помощью макросредств. Для рабо-

ты с макросредствами в EXTED имеется команда MACRO. Далее выдается специальное меню, позволяющее указать создание (CREATE), чтение с диска (GET), вставку (INSERT) текста в макросы, отображение списка (LIST) имеющихся макросов, сохранение (SAVE) макросов в файле.

Ядро. Основа любой прикладной системы, выполненной на базе BOC1810, — ядро, основными функциями которого являются: диспетчерирование; управление доступом к системным ресурсам; обеспечение связей между процессами; обеспечение возможности отклика системы на внешние события.

Ядро обслуживает:

1) задачи — активные объекты, выполняющие работу системы;

2) задания — среда, в которой функционируют задачи, среда состоит из задач, используемых ими объектов, каталога объектов и пула памяти;

3) сегменты — участки памяти, являющиеся средством для общения и запоминания данных задачами;

4) почтовые ящики, посредством которых задачи посылают (принимают) другие объекты;

5) семафоры, позволяющие задачам обмениваться сигналами;

6) регионы — хранят особые наборы разделяемых данных;

7) дополнительные объекты — объекты, определяющие новые типы объектов;

8) составные объекты — объекты новых типов, определенные дополнительными объектами.

Ядро предоставляет набор операций, называемых системными вызовами, которые могут использоваться задачами для работы с объектами.

Далее кратко описываются основные системные вызовы ядра BOC1810.

Вызовы для заданий:

RQ\$CREATE\$JOB — создание задания;

RQ\$DELETE\$JOB — удаление задания.

Вызовы для задач:

RQ\$CREATE\$TASK — создание задачи;

RQ\$DELETE\$TASK — удаление задачи, не являющейся задачей обработки прерывания;

- RQ\$GET\$PRIORITY — получение приоритета задачи;
 RQ\$RESUME\$TASK — уменьшение глубины приостанова задачи на 1; возобновление задачи, если глубина приостанова равна 0;
 RQ\$SET\$PRIORITY — установ приоритета задачи;
 RQ\$SLEEP — помещение задачи в состояние ожидания на указанное время;
 RQ\$SUSPEND\$TASK — увеличение глубины приостанова задачи; приостанов задачи, если она еще не была приостановлена.

Вызовы для почтовых ящиков:

- RQ\$CREATE\$MAILBOX — создание почтового ящика;
 RQ\$DELETE\$MAILBOX — удаление почтового ящика;
 RQ\$RECEIVE\$MESSAGE -- прием объекта из почтового ящика,
 RQ\$SEND\$MESSAGE — посылка объекта в почтовый ящик.

Вызовы для семафоров:

- RQ\$CREATE\$SEMAPHORE — создание семафора;
 RQ\$DELETE\$SEMAPHORE — удаление семафора;
 RQ\$RECEIVE\$UNITS — запрос элементов из семафора;
 RQ\$SEND\$UNITS — добавление элементов в семафор.

Вызовы для сегментов и пулов памяти:

- RQ\$CREATE\$SEGMENT — создание сегмента;
 RQ\$DELETE\$SEGMENT — возврат сегмента в пул памяти;
 RQ\$GET\$POOL\$ATTRIBUTES — получение характеристик пула памяти;
 RQ\$GET\$SIZE — получение размера сегмента.

Вызовы для драйверов особых ситуаций:

- RQ\$CATALOG\$OBJECT — помещение объекта в каталог объектов;
 RQ\$GET\$TYPE — получение типа объекта;
 RQ\$UNCATALOG\$OBJECT — удаление объекта из каталога объектов.

Вызовы для драйверов особых ситуаций:

- RQ\$SET\$EXCEPTION\$HANDLER — определение драйвера особых ситуаций и его атрибутов;

RQ\$GET\$EXCEPTION\$HANDLER — получение адреса драйвера особых ситуаций и его атрибутов.

Вызовы для драйверов прерываний, задач обработки прерываний и уровней прерываний:

- RQ\$DISABLE** — запрет уровня прерываний;
RQ\$ENABLE — разрешение уровня прерываний;
RQ\$END\$INIT\$TASK — информирование корневой задачи о завершении процесса инициализации;
RQ\$ENTER\$INTERRUPT — установ базового адреса сегмента для вызываемого драйвера прерывания;
RQ\$EXIT\$INTERRUPT — посылка в аппаратуру сигнала окончания прерывания;
RQ\$RESET\$INTERRUPT — отмена назначения драйвера прерываний уровню прерывания и, если возможно, удаление задачи обработки прерываний для данного уровня;
RQ\$SET\$INTERRUPT — назначение драйвера прерываний и, если необходимо, задачи обработки прерываний;
RQ\$SIGNAL\$INTERRUPT — запуск драйверами прерываний задач обработки прерываний;
RQ\$WAIT\$INTERRUPT — помещение задачи обработки прерываний в ожидании запуска ее драйвером прерывания.

Вызовы для составных объектов:

- RQ\$ALTER\$COMPOSITE** — замена компонентов составного объекта;
RQ\$CREATE\$COMPOSITE — создание составного объекта;
RQ\$DELETE\$COMPOSITE — удаление составного объекта;
RQ\$INSPECT\$COMPOSITE — получение списка компонентов, входящих в составной объект.

Вызовы для дополнительных типов:

- RQ\$CREATE\$EXTENSION** — создание нового типа;
RQ\$DELETE\$EXTENSION — удаление дополнительного типа и всех составных объектов этого типа.

Вызовы для расширений БОС1810:

- RQ\$SET\$OS\$EXTENSION** — вставка — удаление адреса процедуры входа в таблицу векторов прерывания;
RQ\$SIGNAL\$EXTENSION — сигнализация появления особой ситуации.

Вызовы для регионов:

- RQ\$ACCEPT\$CONTROL — получение региона в том случае, если он доступен немедленно;
- RQ\$CREATE\$REGION — создание региона;
- RQ\$DELETE\$REGION — удаление региона;
- RQ\$RECEIVE\$REGION — получение региона с ожиданием его доступности;
- RQ\$SEND\$CONTROL — передача управления следующей задаче, ожидающей регион.

2.4. Базовое прикладное программное обеспечение

В данном разделе рассматриваются только некоторые из прикладных программ, в первую очередь уже разработанные и широко распространенные, апробированные массовым использованием.

Большое число базовых прикладных программ доступно через фонды алгоритмов и программ.

Базы данных и информационно-поисковые системы. Для СМ 1800 имеется ряд базовых программных пакетов, полезных для автоматизации учрежденческих работ.

1. Пакет ПОДГОТОВКА ДАННЫХ позволяет производить регистрацию и статистическую обработку данных в документе произвольного формата. Статистическая обработка включает в себя выборку записей, их группирование по признакам, вычисление средних, сумм, отклонений. Пакет работает как автономное программное средство. Развитие этого пакета работает под управлением ОС1800; в нем помимо перечисленных возможностей имеются логическая обработка, условные операторы, средства вычисления элементарных функций, минимальных и максимальных значений.

2. Пакет ДИАЛОГ-М — простая информационно-поисковая система, обеспечивающая возможность создания базы данных, поиска и отбора документов, а также средства преобразования и вывода результатов поиска. Структура обрабатываемых документов легко изменяется. Пакет работает под управлением операционной системы МикроДОС.

3. Пакет БнД-80 реализует реляционную базу данных, структура которых гибко определяется. Возможен поиск по одному или нескольким связанным файлам, легко организуется вывод найденной информации как отчетов и таблиц. Пакет включает эффективный командный язык.

Операционной средой для этого пакета является МикроДОС.

4. Пакет СУБД-СМ-1800 поддерживает сетевые структуры с неограниченным числом связей между данными, содержит язык описания схем и подсхем баз данных, обеспечивает различные виды манипулирования данными.

5. Пакет КАЛЕНДАРЬ предназначен для автоматизации исполнительской дисциплины. Работает в диалоговом режиме под управлением ОС1800.

6. Пакет ЭКОНОМИКА [8] — универсальное интегрированное программное средство для обработки экономической информации. Он состоит из ядра, утилит и проблемно-ориентированных прикладных программ. Ядро выполняет функции операционной системы разделения времени (до 8 пользователей), управления ресурсами СМ1800, а также базами данных. Прикладные программы ориентированы на обработку документов различного формата. Например, программа ДЕЛО позволяет вести обработку цифровых и текстовых документов достаточно произвольной табличной формы с помощью специального простого языка описания данных. Достоинство этой программы — развитый аппарат включения библиотечных элементов. Для обработки различных карточек имеется программа МИСС (малая информационно-справочная система).

Обработка текстов. Одна из важных областей применения микроЭВМ СМ 1800 — обработка текстовой информации. Помимо рассмотренных ранее обычных редакторов текстов, работающих в строчном и экранном режимах, для этой цели применяются следующие пакеты.

1. Пакет ТЕКСТ позволяет создавать и (или) редактировать текстовый документ в экранном режиме. При этом возможны определение горизонтального и вертикального формата документа, автоматическое формирование строк из слов, нумерация листов. Упрощены процессы вставки и удаления текста, перемещения и объединения текстов. Для вычислений в пакете имеются встроенные функции счета (+, —, %). Имеется возможность управлять форматом печати документа, готовить индекс на основе ключевых слов. Пакет работает автономно, без использования какой-либо операционной системы. Развитием этого пакета является пакет ТЕКСТ-ОС, работающий под управлением ОС1800.

2. Универсальный экраный текстовый процессор СУПЕРТЕКСТ работает под управлением МикроДОС и позволяет готовить текстовые документы на русском и английском языках. Обеспечивается одновременная работа с несколькими файлами (до 11), двухоконное отображение фрагментов текстов одного или нескольких документов, гибкая настройка на тип применяемого терминала.

3. Пакет программ для подготовки текстовой документации работает под управлением ДОС1800. В пакет входят:

DOCFOR — форматор документов;

DOCCNT — программа вывода оглавления документа;

DOCIND — программа вывода предметного указателя;

DOCPRT — сервисная программа печати ранее сформатированных текстовых документов.

Занесение текстовых документов осуществляется с помощью входящего в состав ДОС1800 редактора текстов EDIT80. Для управления структурой выходного документа пользователь имеет возможность вносить во входной файл так называемые управляющие термы, начинающиеся со знака &. Имеются управляющие термы для окончания абзаца (& C), определения начала (&) и конца (&)) заголовка, выделения текста яркостью (& Я), разрядкой (& P) и разрядкой с подчеркиванием (&—), определения термина (& T), условного (& У) и безусловного (& Л) перевода листа и др.

В процессе работы программы DOCFOR автоматически выравнивается текст по левой и правой границе, осуществляется обработка заголовков и составление оглавления (содержания) документа, формируется предметный указатель. Пакет ориентирован на подготовку программной текстовой документации, соответствующей стандартам ЕСПД, в том числе документов большого объема, разбитых для простоты на несколько файлов, которые, возможно, хранятся на различных дискеттах.

Библиотеки подпрограммы. Для разработки прикладного программного обеспечения важным вопросом является реализация в виде набора подпрограмм основных часто используемых операций, таких, как арифметика, элементарные функции, обработка текстовых строк и т. д.

В операционной среде системы ДОС1800 для этой цели имеется две библиотеки подпрограмм: арифмети-

ческих подпрограмм БАП; программ общего назначения БИПОН [9].

Рассмотрим БИПОН, так как БАП функционально совместима с частью БИПОН, предназначенной для выполнения арифметики над числами с плавающей запятой. БИПОН включает в себя 9 частей:

1) базисные подпрограммы, которые реализуют быстрые строковые операции, а также двоичную и десятичную целую арифметику (без обработки ошибок);

2) двоичную целую арифметику над знаковыми и беззнаковыми числами;

3) арифметику с плавающей запятой;

4) десятичную арифметику;

5) функции обработки строк (поиска, сравнения, исключения и вставки подстрок, табличных преобразований, просмотра входных строк и форматирования выходных строк);

6) подпрограммы преобразования чисел из одной формы представления в другую, включая разбор входных числовых строк и форматирование выходных десятичных строк;

7) трансцендентные функции, вычисляющие тригонометрические, экспоненциальные и другие функции;

8) статистику, содержащую подпрограммы для получения средних значений, дисперсии, среднеквадратичных отклонений, коэффициентов корреляции и др.;

9) подпрограммы ПИД — регулирования.

Все подпрограммы БИПОН реентерабельны, т. е. повторно-входимые. Это дает возможность применять их как в пользовательских программах, работающих под управлением ДОС1800, так и в программах реального времени, например для МОС РВ. Разрядность вычислений в подпрограммах различная, пользователь определяет ее исходя из соображений точности и быстродействия.

Подпрограммы БАП и БИПОН поставляются как наборы библиотек в формате ДОС1800. Их связь с пользовательскими программами — путем прикомпоновки соответствующих библиотек с помощью компоновщика LINK, входящего в состав ДОС1800.

Построение комплексов моделей семейства СМ 1800 и их применение



В гл. 1 и 2 дается описание технических и программных средств семейства микроЭВМ СМ 1800. В настоящей главе рассматриваются возможности формирования комплексов из средств, входящих в состав моделей СМ 1800, и способы поставки пользователям. Построение микроЭВМ СМ 1800 модульное, поэтому процессы поставки комплектующих изделий, изготовления, сборки, приемки готовых изделий связаны с различными трудностями. Например, для переменного состава модулей необходимо заказывать комплектующие изделия и материалы. При изготовлении и сборке используется переменный состав технологического оборудования; при приемке изделий — различное контрольное оборудование и т. д. С одной стороны, процессы изготовления и поставки изделий значительно упрощаются при ограниченном числе типовых комплексов. Однако, с другой стороны, использование типовых комплексов часто приводит к избыточности поставляемых средств, вынужденному ограничению гибкости структуры, а также увеличению стоимости. Поэтому используются различные комбинированные способы изготовления, формирования и поставки комплексов семейства СМ 1800.

3.1. Формирование комплексов

Семейство микроЭВМ СМ 1800 включает в себя 8-разрядные модели СМ 1800 (рис. 3.1) и 16-разрядные модели СМ 1800 (рис. 3.2, 3.3, 3.4).

На все функциональные модули, базовые конструкторские элементы и операционные системы семейства микроЭВМ СМ 1800 разрабатываются собственные комплекты документации, обеспечивающие возможность их автономной поставки заводами-изготовителями. Такая поставка изделий обеспечивает безызыточную комплектацию систем управления, дает возможность последовательно наращивать их и расширять, формировать комплексы по спецификации заказчика. Существует три способа поставки заводам технических средств СМ 1800: типовые комплексы; специфицированные комплексы; отдельные компоненты, называемые «россыпью».

Типовой комплекс семейства СМ 1800 пред-



Рис. 3.1. МикроЭВМ СМ 1800



Рис. 3.2. МикроЭВМ СМ 1810 настольная



Рис. 3.3. МикроЭВМ СМ 1810, тумбовое исполнение



Рис. 3.4. МикроЭВМ СМ 1810, стоечное исполнение

ставляет собой набор модулей, устройств, конструкторских элементов, систем электропитания, комплектов программного обеспечения, документации; используется для выполнения функций управления, определяемых возможностями и составом перечисленных компонентов микроЭВМ. Типовой комплекс является поставляемой единицей. Состав комплексов может быть различным. В табл. 3.1 приводится состав комплексов СМ 1800 на этапе выпуска установочной партии.

Из приведенной таблицы видно, что комплексы СМ 1800.10 — СМ 1810.40 не имеют в своем составе периферийных устройств и комплекта программного обеспечения. Эти комплексы могут являться основой при формировании управляющих вычислительных комплексов специфицированных (УВКС) или встраиваться в оборудование и системы.

Комплекс СМ 1800.11 представляет собой настольный вариант микроЭВМ с периферией и комплектом программного обеспечения; может использоваться как в промышленной, так и непромышленной сферах в виде АРМ или терминалов. Комплекс СМ 1810.21 используется в системах сбора и обработки массивов информации в сетях ЭВМ; предусматривается расширение его дополнительными терминалами и дисковыми накопителями типа «Винчестер». Комплекс СМ 1810.31 предназначен для построения систем управления объектами; может расширяться модулями ввода — вывода и УСО. Ком-

Таблица 3.1

Наименование и обозначение (шифр)	Комплексы СМ 1810							
	10	20	30	40	11	21	31	41
Модуль центрального процессора СМ 1810.2204	1	1	1	1	1	1	1	1
Модуль системного контроля СМ 1810.2005	1	1	1	1	1	1	1	1
Модуль центрального процессора СМ 1800.2202	—	—	—	—	1	1	1	1
Блок питания В253.01	1	1	1	1	1	1	1	1
Блок питания В253.03	—	—	1	1	—	—	1	1
Модуль связи с ИРПС многоканальный СМ 1800.4106	—	—	—	—	1	1	1	1
Контроллер НГМД СМ 1810	—	—	—	—	1	1	1	1
Накопитель на гибком магнитном диске СМ 5640	—	—	—	—	2	2	2	2
Блок приборный НГМД	—	—	—	—	—	—	1	1
Блок приборный БПР-1	1	—	—	—	1	—	—	—
Блок монтажный БМ-1	—	1	—	—	—	1	—	—
Блок монтажный БМ-2	—	—	1	—	—	—	1	—
Блок монтажный БМ-3	—	—	—	1	—	—	—	1
Устройство печатающее алфавитно-цифровое СМ 6329.01 (ГДР)	—	—	—	—	1	1	—	—
Устройство печатающее алфавитно-цифровое СМ 6317 (ГДР)	—	—	—	—	—	—	1	—
Устройство печатающее алфавитно-цифровое СМ 6309 (ГДР)	—	—	—	—	—	—	—	1
Модуль оперативный запоминающий СМ 1810.3515	—	—	—	—	—	—	1	—
Модуль оперативный запоминающий СМ 1810.3516.03	—	—	—	—	—	—	—	1
Видеотерминал алфавитно-цифровой СМ 7209 (ПНР)	—	—	—	—	1	1	1	1
Стойка СМ 1810.0102	—	—	—	—	—	—	1	1
Стойка СМ 1810.0103	—	—	—	—	—	1	—	—
Комплект программного обеспечения	—	—	—	—	1	1	1	1
Комплект эксплуатационной документации	1	1	1	1	1	1	1	1
Комплект запасных частей	1	1	1	1	1	1	1	1
Комплект монтажных частей	1	1	1	1	1	1	1	1
Комплект инструментов и принадлежностей	1	1	1	1	1	1	1	1

плекс СМ 1810.41 включает в себя системное ОЗУ емкостью до 4 М байт; может расширяться устройствами ввода — вывода и дисковыми накопителями типа «Винчестер». Основное назначение — сбор и обработка больших массивов оперативной информации в процессах управления или научного эксперимента.

В процессе развития производства число типовых комплексов может увеличиваться. Так, в составе микроЭВМ СМ 1803 имеется десять комплексов: СМ 1803.01 — СМ 1803.10, а в составе СМ 1804 — шесть: СМ 1804.01 — СМ 1804.06.

Таким образом, заказывая определенный типовой комплекс, пользователь получает набор средств СМ 1800 соответствующего состава.

Специфицированный комплекс набирается пользователем из имеющегося на заводе определенного набора средств СМ 1800 (конфигуратора), предварительно согласовывает состав с заводом-изготовителем под заданные системные требования. Такой способ поставки в наибольшей степени удовлетворяет потребительский спрос, исключает избыточность и позволяет максимально использовать системные особенности микроЭВМ. Существенный недостаток этого способа поставки средств СМ 1800 — увеличение времени цикла «заказ — поставка», связанного с индивидуальными заказами комплектующих изделий для каждого комплекса. Несмотря на указанный недостаток, способ поставки средств СМ 1800 в виде специфицированных комплексов наиболее прогрессивен.

Отдельные компоненты микроЭВМ — это модули, конструктивные элементы, устройства и т. д. Поставка в виде отдельных компонентов применяется для комплектации систем управления, где средства СМ 1800 встраиваются в приборы, оборудование или устройства. Кроме того, этот способ поставки используется для расширения типовых комплексов, а при необходимости и специфицированных.

Описанные способы поставки распространяются на все модели семейства СМ 1800: СМ 1803; СМ 1804; СМ 1810; СМ 1814. Комбинация трех описанных способов обеспечивает гибкое построение систем управления в различных областях применения управляющей вычислительной техники.

3.2. Примеры построения комплексов

Проиллюстрируем формирование комплексов примерами.

Пусть требуется сформировать на базе микроЭВМ семейства СМ 1800 комплекс, предложенный для обработки деловой информации в отделе технического контроля завода. Исходными данными для спецификации альтернативных управляющих вычислительных комплексов (УВК) являются: состав типовых комплексов и модулей расширения из соответствующих технических условий (ТУ) на модели микроЭВМ; перечень модулей, конструктивных элементов и устройств, входящих в ограничительный перечень для формирования управляющих вычислительных комплексов специфицированных (УВКС).

Проектируемый УВК реализуется на моделях семейства микроЭВМ СМ 1800. Поэтому, воспользовавшись ТУ на эти модели, можно выбрать микроЭВМ, которая наиболее полно удовлетворяет требованиям проектируемого комплекса.

Для удобства заказа и организации изготовления требуемой микроЭВМ целесообразно представить ее в виде типового комплекса и дополнительных модулей из перечня УВКС. На этом заканчивается этап системной проработки выбора микроЭВМ. Допустим, что на этом этапе определено, что проектируемый УВКС должен состоять из типового комплекса СМ 1810.11 (см. табл. 3.1.) и одного модуля дискретного ввода на 16 каналов СМ 1800.9302 [2]. Способ и условия установки этого модуля в комплексе, где имеется одно свободное место, определяются из перечня УВКС. Предполагается, что такой комплекс должен обеспечить функции ввода — вывода и обработки информации, а модуль дискретного ввода — связь со стендом, контролирующим готовое изделие.

Таким образом, УВКС на базе СМ 1810 состоит из следующих компонентов:

Типовой комплекс СМ 1810.11	— 1
Модуль дискретного ввода СМ 1800,9302	— 1
Комплект системной документации на УВКС	— 1

Программное обеспечение входит в комплекс СМ 1810.11 и документацию на УВКС. Таким образом, сформированный УВКС представляет собой настольный приборный вариант СМ 1810 (рис. 3.2) с дисплеем и печатающим устройством, предназначенный для оборудования рабочего места работника ОТК завода.

Рассмотрим еще один пример формирования комплекса, предназначенного для создания локальной системы управления технологическим процессом на производстве. Предположим, что в проектируемую систему должна входить микроЭВМ семейства СМ 1800 со следующими устройствами связи с объектом (УСО):

- 64 канала аналогового ввода среднего (0—5 В) уровня;
- 16 каналов аналогового вывода;
- 16 каналов дискретного ввода;
- 16 каналов дискретного вывода.

Определив модель микроЭВМ и ее состав, представим микроЭВМ в виде одного из типовых комплексов и дополнительного набора модулей. Допустим, что проектируемый УВКС включает в себя типовой комплекс СМ 1810.31 и модульный набор. В свою очередь, модульный набор состоит из модулей дискретного и аналогового ввода — вывода. Поскольку СМ 1810.31 представляет собой стойку, в верхнем ряду которой в блоке монтажном (БМ) установлена базовая ЭВМ, а остальные места БМ свободны, расположим УСО на свободных местах. Для организации дискретного ввода — вывода необходим один модуль МВВД СМ 1800.9302 и два модуля МВД СМ 1800.9303, для расположения которых требуется три места. Целесообразно использовать свободные места в БМ рядом с модулями базовой ЭВМ. Установим на эти свободные места модули дискретного ввода — вывода. Для организации аналогового ввода — вывода необходимы следующие модули:

- Модуль аналогового ввода МВВА-1 (один блок элементов) СМ 1800.9204 — 1
- Модуль источника питания МИП (один блок элементов) СМ 1800.0303 — 1
- Модуль коммутации аналоговых сигналов МКАС-2 (один блок элементов) СМ 1800.8518 — 4
- Модуль аналогового вывода МАВ (два блока элементов) СМ 1800.9202 — 4

Все эти модули устанавливаются в отдельный блок расширения (БР) СМ 1800.0105, который расположен в средней части стойки СМ 1810.31. Для организации физической связи с объектом в нижней части стойки СМ 1810.31 устанавливается типовой кросс, содержащий две рамы по восемь мест. В кросс устанавливаются:

- Модуль аналогового питания — МАП (четыре блока элементов) СМ 1800.0302 — 1

Модуль нормализации аналоговых сигналов —	
МНАС-2 (один блок элементов) СМ 1800.9211.04	— 8
Колодки с клеммами (для связи с МАВ)	— 4

Общий вид комплекса приведен на рис. 3.4.

В процессе эксплуатации комплексов может возникнуть необходимость в расширении типовых или специфицированных комплексов введением дополнительных модулей и устройств. Учитывая тот факт, что основные принципы построения семейства микроЭВМ СМ 1800 предусматривают модульное построение всей системы и изготовление полного комплекта документации на каждый модуль, а также определение цены, поставку отдельных модулей и устройств планирует каждый завод-изготовитель. Условия подключения отдельных компонентов микроЭВМ предусматриваются в соответствующих ТУ.

В предыдущих примерах приведено построение комплексов семейства СМ 1800 из серийно выпускаемых модулей, блоков и элементов конструкций. Рассмотрим построение такого комплекса, в котором дополнительно к заводским компонентам микроЭВМ требуется добавить модуль, разработанный пользователем. Концепция построения семейства СМ 1800 как открытой системы такова, что она допускает расширение технических средств с условием соблюдения стандартов СМ ЭВМ на конструкцию, интерфейсы, программное обеспечение и др. При этом, естественно, снимаются гарантии завода-изготовителя за работоспособность комплексов. Разработка программного обеспечения для данного модуля возлагается также на пользователя. При установке модуля следует руководствоваться основными документами и ТУ на комплекс и входящие в них модули. Построение комбинированных комплексов, как правило, доступно отдельным лицам или организациям, профессионально занимающимся разработкой средств вычислительной техники.

Сроки поставки заказываемых комплексов различны для типовых УВК, УВКС и отдельных модулей. Они определены типовыми положениями на поставку микроЭВМ семейства СМ 1800. Прежде чем сделать заказ на УВКС, следует ознакомиться с вариантами, реализованными заводом; если среди этих вариантов найдется такой, который устраивает потребителя, то срок поставки может быть сокращен.

Техническое обслуживание комплексов осуществляется организацией СОЮЗЭВМСЕРВИС, которая имеет

разветвленную сеть по всей территории СССР. Основанием для обслуживания является типовый договор, который оформляется между пользователем и СОЮЗЭВМ-СЕРВИС. В период гарантийного срока обслуживание осуществляется заводом-изготовителем комплекса.

Номенклатура изделий семейства микроЭВМ СМ 1800 постоянно расширяется, появляются интеллектуальные модули, которые могут заменить, например, несколько модулей, разработанных ранее и выпускающихся другими заводами. Поэтому оценку и выбор технических средств, закладываемых в проекты систем, следует увязывать с перспективами развития семейства СМ 1800. Построение современных систем управления только на базе семейства микроЭВМ СМ 1800 значительно упрощает процессы обучения персонала, эксплуатационного обслуживания, разработки программного обеспечения и позволяет получить наибольший эффект от внедрения вычислительных средств.

3.3. Области применения

Одной из основных форм представления информации, используемой в процессе разработок моделей семейства СМ 1800, являются системные требования. Формирование их связано со сбором, анализом и каталогизацией данных о планируемых областях применения разрабатываемых микроЭВМ. Поэтому начало разработок семейства СМ 1800 связано с созданием банка системных применений (СП), который постоянно дополняется и анализируется. Источниками для формирования и расширения БСП являются: результаты проводимого анкетирования организаций-разработчиков систем, анализ отечественных и зарубежных разработок, материалы статистики, отчеты патентных исследований, литературные данные, результаты научно-исследовательских работ.

Для представления системных требований выбраны следующие классификационные признаки: области применения; выполняемые функции; архитектурные признаки; виды обрабатываемой продукции.

Микропроцессорные системы управления применяются в различных областях — промышленность, экономика, научные исследования, непромышленная сфера. Более детальная классификация охватывает десятки отраслей, так или иначе относящихся к указанным четырем областям применений. Системные проработки и опыт эксплуа-

тации 8-разрядных моделей микроЭВМ СМ 1800 позволили определить и уточнить ряд областей применения, в которых эти модели успешно функционируют, обеспечивая получение экономического эффекта, — управление инерционными процессами в машиностроении, стендовый контроль электронно-технологического оборудования, автоматизация обработки информации о подвижном составе на железнодорожном транспорте, сбор и обработка планово-экономической информации, обработка картографической информации, проектирование и отладка микропроцессорных систем и др. Создание СМ 1810 и СМ 1814 позволило расширить применение семейства СМ 1800 в указанных областях, а также определить новые применения в области науки и техники.

Как показал опыт, при формировании системных требований целесообразно использовать классификацию систем по выполняемым функциям. При этом классификационных признаков становится существенно меньше, и кроме того, им могут быть поставлены в соответствие в значительной мере совпадающие для каждой выполняемой функции вектора системных требований. Так, для реализованных систем можно назвать отдельно выполняемые функции: контроль, измерение, сбор данных, регулирование, обработку изображения и т. д.

К архитектурным признакам, характеризующим систему управления, относятся: структура построения средств, магистрали микропроцессорных наборов, внутрисистемные и внешние интерфейсы, карты распределения памяти и портов, элементы программного обеспечения, протоколы связи, тип кодирования и др. Поскольку развитие микропроцессорных систем идет в направлении приближения средств обработки информации к месту ее возникновения, как правило, структура построения таких систем рассредоточенная и ориентирована на параллельную обработку информации. Отличительной особенностью построения рассредоточенных микропроцессорных систем является свойство отражения во внутренней структуре микроЭВМ, используемых в этих системах, топологии объекта управления. Территориально рассредоточенным системам управления соответствует многоуровневая организация микроЭВМ семейства СМ 1800, а также многомашинные комплексы, объединенные в локальные сети.

Рассмотрим основные направления применений последних разработок семейства СМ 1800: СМ 1810 и СМ 1814.

1. Машиностроение. Существенным недостатком 8-разрядных моделей СМ 1800 использования в этой области явилась ограниченная производительность, равная 80—100 тыс. оп./с. Для 16-разрядных микроЭВМ СМ 1810 и СМ 1814 значение производительности в однопроцессорном режиме составляет более 500 тыс. оп./с. Такая производительность, наличие технических средств и соответствующего программного обеспечения позволяют применять указанные микроЭВМ для управления в реальном масштабе времени технологическим оборудованием, конвейерными линиями, гибкими производственными модулями, роботами и манипуляторами, а также использовать в составе гибких автоматизированных производств. Наличие средств связи и специального конструктивного исполнения СМ 1814 позволяют создавать локальные технологические сети для машиностроительных производств, а также для производств с ограниченным доступом обслуживающего персонала. Структурная гибкость и наличие разнообразных модулей ввода — вывода обеспечивают применение микроЭВМ в виде встраиваемой, локальной и распределенной системы управления.

Для расширения областей применения моделей семейства СМ 1800 в машиностроении выполняются разработки для построения управляющих комплексов гибких производственных систем — модулей управления приводами, модулей активного контроля, модулей управления промышленными мониторами и др.

МикроЭВМ СМ 1810 может использоваться в качестве базовой ЭВМ в автоматизированных рабочих местах (АРМ). Разработка АРМ, ориентированного на использование в машиностроении, известна под шифром «Автограф 840». Основное назначение АРМ — автоматизированное проектирование, ведение библиотек изделий, разработка конструкторской документации, получение носителей для станков с ЧПУ, разработка текстовой и технологической документации. В составе АРМ совместно с типовым комплексом СМ 1810.50 используется графический растровый дисплей «Автограф 841», устройство получения твердой копии экрана «Автограф 842», устройство ввода графики, печатающее устройство, графопостроители формата А3 и А0 и устройство ввода — вывода перфоленточное.

Типовой комплекс СМ 1810.50 включает в себя следующие компоненты:

- модуль системного контроля (МСК);

- модуль МЦП-16;
- модуль ОЗУ емкостью 1 М байт;
- комбинированный контроллер гибкого диска и диска типа «Винчестер»;
- модуль ввода — вывода для подключения периферии;
- операционную систему реального времени — БОС1810.

Конструктивно СМ 1810.50 выполняется во встраиваемом варианте исполнения. Прикладное программное обеспечение состоит из пакета прикладных программ (ППП) «Автограф 845», которые предоставляет в распоряжение проектировщика практические средства, позволяющие разрабатывать и воспроизводить на экране дисплея схемы и общий вид изделия, выполнять подробные чертежи каждой детали, входящей в изделие. Имеется возможность доступа к ранее выполненным чертежам, хранящимся в библиотеке, а также использования различных нормализованных компонентов, таких, как винты, гайки, подшипники и т. д.

Создание АРМ «Автограф 840» на базе микроЭВМ СМ 1810.50 позволяет получить сравнительно недорогое доступное средство автоматизации процессов проектирования в машиностроении.

2. Автоматизированные системы управления технологическим процессом (АСУ ТП). Для семейства микроЭВМ СМ 1800 традиционно применение в АСУ ТП. Широкая номенклатура модулей ввода — вывода, наличие специальных исполнений для промышленных условий, а также развитое индустриальное программное обеспечение реального времени позволяют создавать на базе СМ 1810 и СМ 1814 системы управления технологическими процессами в различных производствах. Примером может служить разрабатываемая система управления технологическими процессами с программным пакетом, именуемым СКАТ 1810.

Система предназначена для использования в таких энергоемких отраслях промышленности, как химическая, металлургическая, горнорудная, строительных материалов, в которых совершенствование технологии дает значительный эффект, в первую очередь за счет снижения затрат энергосырьевых ресурсов. Система СКАТ 1810 базируется на современных достижениях теории автоматического регулирования:

- многосвязной оптимальной стабилизации;

- иерархической организации отношений между управлением заданием и регулированием в отклонениях;
- инвариантном и автономном управлении;
- контроле нарушения адекватности текущего состояния объекта и его линеаризованного описания при значительных отклонениях технологического режима.

В состав системы СКАТ1810 входят инструментальные средства, позволяющие в форме простого диалога ввести данные о конфигурации объекта, терминологическое описание и значения параметров, провести «аналитическое» конструирование оптимальных регуляторов по желаемому критерию в интегральной квадратичной форме, а также создать цветные графические изображения на растровом видеоконтрольном устройстве. Например, возможно отображение мнемосхемы автоматизированного участка технологической линии с последующей динамической цветной кодировкой состояния ее составных элементов в ходе управления производственными процессами. Отмеченные особенности системы СКАТ 1810 обеспечивают значительное повышение качества регулирования технологических процессов по сравнению с традиционными одноконтурными системами ПИД-регулирова-

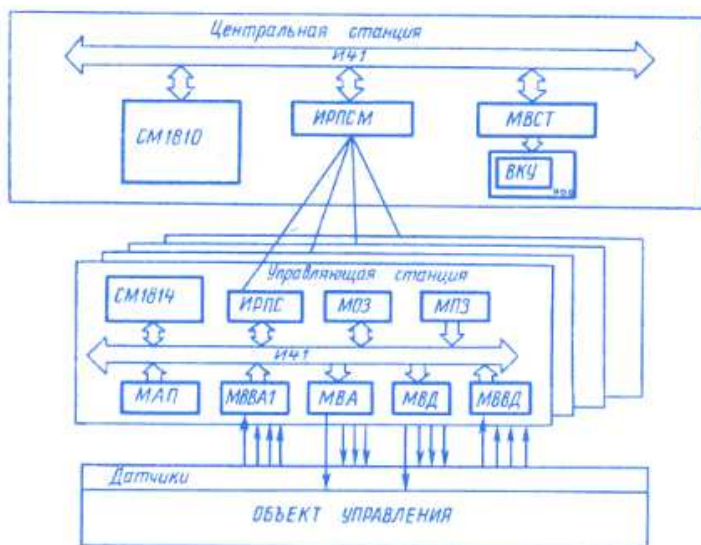


Рис. 3.5. Структурная схема комплекса СМ 1810 для функционирования пакета СКАТ 1810

ния, синтезируемыми на основе опыта и интуиции технологов и тиражируемыми в отечественных и зарубежных разработках.

Структурная схема основной конфигурации распределенного комплекса технических средств для функционирования системы СКАТ1810 приведена на рис. 3.5. Представлена децентрализованная структура из нескольких управляющих станций (УС) на базе СМ 1814, осуществляющих многосвязное регулирование по заданию с центральной станции (ЦС). Единая для всего комплекса (ЦС) (на базе СМ 1810) выполняет функции блока управления вычислительным комплексом, центрального блока обработки и отображения данных и инструментального синтеза многомерных законов регулирования всех УС. Общая задача управления, таким образом, распределяется на две: регулирования (стабилизации) в отклонениях («в малом») от некоторого задания; формирования этого задания (стабилизация «в большом»). Если стабилизация в малом повышает качество продукта и, как следствие, дает прямую экономию энергии и сырья, то стабилизация в большом повышает эффективность использования оборудования за счет сокращения числа переключений, переход с режима на режим, пусков и остановов.

Разрабатываемая система СКАТ1810 представляет новое поколение регулирующих систем, обеспечивающих высокое динамическое качество стабилизации независимо от изменения свойств объекта регулирования и условий его работы.

3. Организационные системы. Наиболее массовым применением СМ 1810 являются организационные системы, которые могут быть локальными и рассредоточенными. Они используются как самостоятельные ветви в комплексных системах АСУ ТП, ГАП, а также в различных непромышленных системах сбора и обработки информации. Выполняемые функции заключаются в планировании и контроле исполнения поручений, сборе и обработке экономической и статистической информации, передаче и представлении информации в требуемой форме, составлении и корректировке отчетов, ведении баз данных и др. Состав технических средств определяется решаемыми задачами и предусматривает компоновку в виде настольного варианта или тумбы. Отличительными признаками комплексов, используемых в организационных системах, являются: ОЗУ до 1 М байт; выходы на каналы связи

через модули ИРПС, МСМ, МСТ, МС; внешние запоминающие устройства типа СМ 5504, СМ 5505, СМ 5408, СМ 5514; дисплеи СМ 7209, СМ 7222, ВТА 2000.15М; печатающие устройства растровые и типа «Ромашка». Операционные системы БОС1810, Микрос-86 или МДОС-1810 обеспечивают работу большого числа разработанных прототипных пакетов прикладных программ, ориентированных на обработку текстовой информации, вычислений, статистических данных, деловых игр и т. д.

Среди пакетов прикладных программ имеются пакеты по самообучению пользователей, которые позволяют значительно сократить сроки внедрения организационных систем. Широкое внедрение этих систем дает возможность сэкономить людские и материальные ресурсы, а также повысить эффективность функционирования управляющих систем.

4. Системы для прогнозирования и анализа разработок полезных ископаемых. Возможность организации в СМ 1810 мультипроцессорного режима работы (до 16 процессоров), наличие дисков типа «Винчестер» емкостью до 160 М байт, оперативной памяти свыше 4 М байт, БИС сопроцессора в МЦП-16; средств для отображения алфавитно-цифровой и графической информации позволяют строить высокопроизводительные недорогие (20—30 тыс. руб.) вычислительные системы для решения задач поля. В этой области преимущественно использовались большие ЭВМ вычислительных центров. Применение микроЭВМ СМ 1810 позволяет решать подобные задачи непосредственно на месте разработок, в партиях, что дает возможность получить экономический эффект за счет оперативности обработки данных.

Из разнообразных средств вычислительной техники наиболее широко в народном хозяйстве применяются персональные ЭВМ. В будущем предполагается реорганизовать систему управления хозяйством и создать эффективную структуру общегосударственных вычислительных ресурсов. В составе СМ 1810 имеются технические и программные средства, позволяющие использовать многочисленные пакеты прикладных программ для персонально-профессиональных ЭВМ. Это позволяет строить однородные рассредоточенные системы управления в промышленной и непромышленной сферах народного хозяйства на единых средствах микроЭВМ семейства СМ 1800 и значительно упростить внедрение и эксплуатацию этих систем.

Указанные области применения микроЭВМ СМ 1810 и СМ 1814 значительно расширяют использование универсальных микроЭВМ семейства СМ 1800, позволяют получить дешевую вычислительную мощность в широком диапазоне применений.

В составе СМ 1810 имеются программные и технические средства, позволяющие создавать комплексы, совместимые с персонально-профессиональной ЭВМ типа ЕС 1841. Основное отличие этих комплексов — возможность работы с несколькими операционными системами, в том числе с ОС реального времени. Эта возможность позволяет эффективно использовать прикладные программы, разработанные как для персонально-персональных ЭВМ, так и для управляющих вычислительных комплексов, работающих под управлением операционной системы реального времени БОС1810.



МикроЭВМ «Искра-226» выпускается серийно с 1981 г. и является одной из первых отечественных микроЭВМ, работающих с языком программирования высокого уровня, рассчитана на персонального пользователя. «Искра-226» широко используется как профессионально-персональная микроЭВМ (ППЭВМ) в различных областях обработки данных. Она позволяет решать задачи управления и автоматизации научных исследований, проведения планово-экономических расчетов, создания информационно-справочных систем и систем обработки информации, организации решения административно-хозяйственных задач. МикроЭВМ может применяться также при подготовке управляющих программ для станков с числовым программным управлением. Схематехническое и программное обеспечение позволяет использовать машину в информационно-вычислительных сетях в качестве интеллектуального терминала ЕС, СМ и других ЭВМ, а также подключать к ним большой набор периферийных устройств. В зависимости от областей применения и вида периферийного оборудования, с которым должна работать микроЭВМ, ее состав может видоизменяться.

4.1. Архитектура микроЭВМ

МикроЭВМ «Искра-226» представляет собой многопроцессорную систему с иерархической структурой (рис. 4.1) [10]. Основу структуры составляет 16-разрядная специализированная центральная микромашина (ЦММ), имеющая права ведущего процессора. Канальная 16-разрядная микромашина (КММ) реализует операции ввода — вывода, а также обеспечивает связь периферийных блоков с оперативной памятью. В КММ имеется возможность реализации кольцевой дисциплины обслуживания мультиплексного канала. В ее функцию входит управление блоком коммутации магистралей (БКМ), который обеспечивает связь между ЦММ и КММ, с ОЗУ и магистралью ввода — вывода, а также параллельную работу ЦММ и КММ. Совокупность ЦММ, КММ, системного ОЗУ, блока коммутации магистралей и блока контроля и

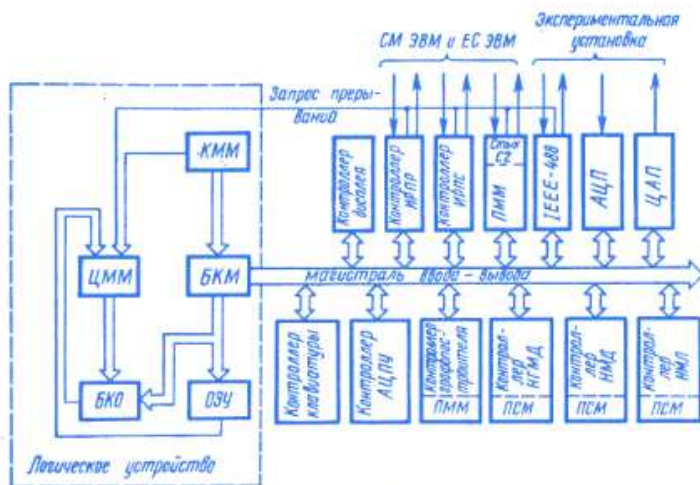


Рис. 4.1. Структурная схема микроЭВМ «Искра-226»

отладки (БКО) образует логическое устройство, автоматически выполняющее последовательности операций преобразования информации и управляющее устройствами ввода — вывода информации в соответствии с используемыми алгоритмами.

Общая магистраль ввода — вывода микроЭВМ состоит из 8-разрядной двунаправленной информационной магистрали, 4-разрядной однонаправленной шины команд, 3-разрядной однонаправленной шины состояний и шин ВЫЗОВ, ОТВЕТ, ЗАПРОС, НАЧАЛЬНАЯ УСТАНОВКА.

На нижнем уровне иерархии находятся периферийные специализированные микромашины (ПСМ), управляющие скоростными внешними устройствами микроЭВМ (ИГМД, ИМЛ, ИМД). Периферийные микромашины (ПММ) ориентированы на реализацию сложных изменяемых в процессе работы алгоритмов и используются в микроЭВМ как сопроцессоры ЦММ. Так как ПММ и ПСМ подключаются к 8-разрядной магистрали ввода — вывода, то их разрядность также составляет 8 бит. Каждая машина имеет внутренние магистрали, разрядности которых могут отличаться от разрядности общей магистрали ввода — вывода микроЭВМ.

МикроЭВМ создавалась в период интенсивного развития элементной базы микроэлектроники, что наложило определенный отпечаток на структуру и идеологию по-

строения отдельных ее узлов. В качестве элементной базы используются интегральные схемы серий К131, К155, К158, К556, К559, К580, К589.

Устройства ввода — вывода (УВВ) подключаются к магистрали ввода — вывода микроЭВМ через блоки интерфейсные функциональные (БИФ), входящие в состав ПСМ и ПММ и служащие для согласования специфических особенностей конкретного УВВ с интерфейсом по магистрали ввода — вывода.

МикроЭВМ «Искра-226» выпускается в нескольких вариантах исполнения (табл. 4.1), отличающихся составом интерфейсных модулей и периферийных устройств [10]. Исполнения 1 и 2 ориентированы на использование в

Таблица 4.1

Наименование устройств и их условное обозначение	Состав модулей и периферийных устройств при исполнении						
	1	2	3	4	6	7	АРМ ТП
Процессор интерпретирующий диалоговый (ПИД «Искра-226») и его подвижные модули	+	+	+	+	+	+	+
Блок светового пера	-	-	-	-	-	+	+
Указатель графической информации типа «джойстик»	-	-	-	-	+	-	-
Контроллер накопителя на гибком магнитном диске БИФ «Искра 015-21»	+	+	+	+	+	+	+
Контроллер накопителя на жестком магнитном диске БИФ «Искра 015-23»	-	+	+	-	-	+	+
Контроллер накопителя на магнитной ленте БИФ «Искра 015-25»	-	-	+	-	-	-	-
Контроллер печатающего устройства и клавиатуры	+	+	+	+	+	+	+
БИФ «Искра 015-30», «Искра 015-31», «Искра 015-33»	-	-	+	-	+	+	-
Контроллер связи, ранг ИРПР БИФ «Искра 015-82»	+	+	+	+	+	+	+
Контроллер связи, СТЫК С2 БИФ «Искра 015-85»	-	-	-	-	-	-	+
Контроллер связи, СТЫК ИРПС БИФ «Искра 015-87»	-	-	+	+	+	+	+
Контроллер графопостроителя БИФ «Искра 015-13»	-	-	-	-	+	+	-
Контроллер приборного интерфейса IEEE-488 БИФ «Искра 015-83»	-	-	-	-	+	+	-

Наименование устройств и их условное обозначение	Состав модулей и периферийных устройств при исполнениих						
	1	2	3	4	6	7	АРМ ТП
АЦП на 32 канала с микроконтроллером БИФ «Искра 015-14»	-	-	-	-	+	+	-
ЦАП БИФ «Искра 015-10»	-	-	-	-	+	+	-
Контроллер телеграфного интерфейса БИФ «Искра 015-36»	-	-	-	-	-	-	-
Контроллер считывателя с перфоленты и перфоратора БИФ «Искра 015-57»	-	-	-	-	-	-	+
Периферийные устройства							
Накопитель на гибких магнитных дисках (2 дисковод на базе ЕС 5074 НРБ или ПЛ 45Д2 ПНР)	+	+	+	+	+	+	+
Накопитель на жестких магнитных дисках (СМ 5400 или ИЗОТ 1370 НРБ)	-	+	+	-	-	+	+
9-дорожечный накопитель на магнитной ленте (на базе СМ 5300 или ИЗОТ 5003)	-	-	+	-	-	-	-
Матричное печатающее устройство: ДЗМ-180 (ПНР), «Роботрон 1154», «Роботрон 1156М» (ГДР)	+	+	+	+	+	+	+
Графопостроитель Н-306	-	-	+	+	+	+	+
Указатель графической информации «джойстик» «Искра 007-50»	-	-	-	-	+	-	-
Фотосчитыватель ФС-1501	-	-	-	-	-	-	+
Перфоратор ленточный ПЛ-150	-	-	-	-	-	-	+

качестве интеллектуального терминала к СМ и ЕС ЭВМ и автоматизацию плановых расчетов; исполнение 3 — на автоматизацию административно-управленческих и планово-экономических задач; исполнение 4 — на автоматизацию научно-исследовательских и инженерных расчетов; исполнения 6 и 7 — на автоматизацию рабочих мест исследователей-экспериментаторов; АРМ ТП — на автоматизацию подготовки, контроля и редактирования управляющих программ для станков с числовым программным управлением и автоматизацию проектирования технологических процессов.

Любая конфигурация машины содержит:

- логическое устройство и набор БИФ, конструктивно размещенных в одном корпусе с дисплеем;
- клавиатуру;
- накопитель на гибких магнитных дисках (НГМД);
- последовательное алфавитно-цифровое печатающее устройство (АЦПУ).

Любое исполнение машины обеспечивает:

- ввод данных и программ с клавиатуры и НГМД;
- редактирование и запуск программ с клавиатуры;
- вывод программ и результатов обработки информации на экран дисплея, АЦПУ, НГМД;
- сопряжение с аппаратурой передачи данных по стандартному интерфейсу СТbК С2 (V.24 или RS 232);
- выполнение операторов программы, записанной в памяти микроЭВМ.

Объем оперативной памяти (исключая объем ОЗУ и ПЗУ в канальных и периферийных процессорах машины) — 128 К байт и 16 К байт ПЗУ. При этом 64 К байт памяти ОЗУ предназначены для размещения интерпретирующей системы и 64 К байт для программ и данных пользователя.

4.2. Основные блоки микроЭВМ

Основной частью микроЭВМ является процессор интерпретирующий диалоговый (ПИД), обеспечивающий ввод — вывод и обработку информации в соответствии с программой, хранимой в его оперативной памяти. Ввод программ и их выполнение осуществляются на уровне микропрограмм, фиксированных или загружаемых в управляющую память. Имеется четыре варианта исполнения ПИД, реализующих различные процедуры в микроЭВМ (табл. 4.2). Среднее быстродействие ПИД составляет $5 \cdot 10^5$ инстр/с. Элементная база — интегральные схемы серий К155, К158, К556, К565, К589. В состав его входят процессор интерпретирующий (ПИ) и клавиатура, подключаемая к ПИ с помощью БИФ. Логическое устройство процессора выполнено в виде одной секции совместно с расширителем ввода — вывода, позволяющим подключать одновременно до семи БИФ.

Консольное устройство ввода — алфавитно-цифровая клавиатура, которая позволяет вводить символы латинского и кирилловского алфавитов (строчные и прописные) и ключевые слова языка программирования БЭЙ-СИК, редактировать текст программ. Клавиатура имеет

Таблица 4.2

Наименование	Количество единиц оборудования, шт., при варианном исполнении ПИД				Примечание
	1	2	3	4	
Устройство логическое (исп. 1)	1	—	—	—	С постоянной управляющей памятью
Устройство логическое (исп. 2)	—	—	1	—	С оперативной управляющей памятью
Устройство логическое (исп. 3)	—	1	—	1	С оперативной управляющей памятью
Накопитель на магнитной ленте кассетный «Искра 005-33»	1	—	1	—	—
Блок питания	1	1	1	1	—
БОСГИ-1920 (исп. 2)	1	1	1	—	Отображение символьной и графической информации
БОСГИ-1920 (исп. 3)	—	—	—	1	Отображение символьной информации на экранах ЭЛТ БОСГИ и ВК
Устройство вентиляции	1	1	1	1	—

32 программируемых функциональных ключа, расширяющих возможности управления ходом выполнения программы. В языке программирования имеется оператор ввода кода нажатой клавиши без индикации его на экране дисплея, что позволяет программировать все клавиши клавиатуры, присваивая им символы любого языка.

Консольное устройство вывода — экран дисплея. Вывод информации на экран осуществляется блоком отображения символьно-графической информации (БОСГИ). Дисплей использует растровый принцип формирования изображения, которое генерируется на экране с частотой 50 Гц, с использованием информации, записанной в ОЗУ дисплея. Каждому элементу памяти ставится в соответствие совокупность точек, отображаемых на экране. БОСГИ позволяет разместить на экране до 1920 (24 строки по 80) символов и (или) отобразить графическую информацию объемом 256 строк по 560 точек в каждой строке. Адреса алфавитно-цифрового и графического устройства различны, хотя они и работают с одной электронно-лучевой трубкой.

Накопитель на гибких магнитных дисках используется как основная внешняя память во всех исполнениях микроЭВМ для записи, хранения и обмена информацией между микроЭВМ и может работать с двумя дискеттами. Емкость каждой стороны дискетты 256 К байт.

Накопитель на сменном кассетном магнитном диске используется в качестве дополнительной внешней памяти с произвольным доступом к информации. Помимо сменного диска устройство имеет встроенный несменный диск. Объем памяти НМД составляет $2 \times 2,5$ М байт.

Накопитель на магнитной ленте используется в качестве дополнительного внешнего устройства хранения информации. Запись информации производится на 9-дорожечную магнитную ленту. НМЛ обеспечивает обмен информацией, записанной на магнитной ленте с ЕС и СМ ЭВМ, и подключается к микроЭВМ через БИФ «Искра 015-25».

Матричное печатающее устройство последовательного типа предназначено для вывода алфавитно-цифровой и символьной информации на печать, осуществляемую в виде матричного раstra 7×7 точек со средней скоростью не менее 40 строк/мин. Максимальное число знаков в строке — 158, ширина рулона — от 100 до 420 мм. Печатаются прописные буквы русского и латинского алфавитов, арифметические знаки, знаки препинания и специальные знаки. Печатающее устройство подсоединяется к магистрали ввода — вывода микроЭВМ через БИФ «Искра 015-33».

Двухкоординатный графопостроитель Н306 предназначен для автоматического вычерчивания графической и символьной информации с шириной поля записи по координате X, равной 300 мм, по координате Y — 200 мм. Максимальная скорость перемещения пишущего узла 75 мм/с, диапазон изменения входной величины по каналу X — от $3 \cdot 10^3$ до $3 \cdot 10^{-2}$ В, а по каналу Y — от $2 \cdot 10^{-3}$ до $2 \cdot 10^{-2}$ В. Графопостроитель подключается к магистрали ввода — вывода микроЭВМ посредством БИФ «Искра 015-13».

Указатель графической информации «Искра 007-50» подключается к микроЭВМ с помощью БИФ «Искра 015-60», в функции которого входит получение графической информации от указателя блока, преобразование ее в двоичный код и передача в микроЭВМ в соответствии с интерфейсом ввода — вывода.

Кроме перечисленных устройств в комплект поставки

микроЭВМ могут входить: накопитель информации на жестких магнитных дисках, различные аналого-цифровые и цифроаналоговые преобразователи. Все варианты исполнения микроЭВМ имеют микропроцессорный контроллер для связи с аппаратурой передачи данных через стандартный СТЫК С2 — БИФ ТК «Искра 015-85».

Ряд функциональных узлов микроЭВМ имеет традиционное схемотехническое и программное обеспечение и используется для организации работы ее внутренних блоков (например, контроллер клавиатуры, БОСГИ, КНГМД и т. д.), поэтому наибольший интерес, с точки зрения понимания общих принципов построения и функционирования машины, представляет рассмотрение вопросов организации ее магистралей ввода — вывода.

4.3. Интерфейс ввода — вывода информации

Обмен информацией между логическим устройством и внешними устройствами машины осуществляется по магистрали ввода — вывода с использованием БИФ. При этом логическое устройство координирует работу всех блоков системы единым алгоритмом; синхронизирует работу всех БИФ, входящих в систему, а также анализирует сигналы неисправностей БИФ, УВВ и процессора.

Основная функция БИФ — сопряжение УВВ с магистралью ввода — вывода для приема и передачи данных между микроЭВМ и УВВ. Кроме того, БИФ может осуществлять контроль принимаемых данных с целью обнаружения ошибок. Информирование логического устройства о готовности данных к обмену может осуществляться как по сигналам прерывания (активный БИФ), так и программными методами (пассивный БИФ). Установка БИФ в активный режим осуществляется по команде процессора.

Любой БИФ содержит узлы для обнаружения как собственных неисправностей, так и неисправностей УВВ, с которым он работает. Выдача сигнала неисправности осуществляется БИФ только в ответ на соответствующий запрос логического устройства.

По отношению к процессору БИФ могут быть: приемником информации; источником информации; приемником-источником информации.

Структура интерфейса представлена на рис. 4.2. Восьмиразрядная информационная магистраль ввода — вывода (МВВ) предназначена для побайтового обмена



Рис. 4.2. Структура интерфейса микроЭВМ

процессору сигнала запроса на обмен информацией по магистрали ввода — вывода; НАЧАЛЬНАЯ УСТАНОВКА — для начальной установки процессором всех БИФ в исходное состояние.

Обмен по информационной магистрали осуществляется с использованием интерфейсных команд (табл. 4.3), передаваемых в БИФ по шинам команд, и интерфейсных состояний, устанавливаемых БИФ на шинах состояний МВВ (табл. 4.4). Начало любого процесса обмена информацией устанавливается процессором путем выдачи команды УСТАНОВИТЬ СВЯЗЬ (УС) на шину кодов команд и установления физического адреса устройства (ФАУ) на информационные магистрали. При выполнении команды БИФ с заданным ФАУ подключается к МВВ, а остальные БИФ функционально отключаются от МВВ. Команда УС не прерывает и не изменяет выполнения устройством ранее принятой команды. В результате выполнения команды подключенный к МВВ БИФ устанавливает на шинах состояния коды интерфейсного состояния (табл. 4.4). Эти коды могут соответствовать разрешенным или запрещенным состояниям. Во втором случае ситуация расценивается как сбой. В дальнейшем на алгоритмах функционирования микроЭВМ коды интерфейсных состояний указываются в соответствии с табл. 4.4. На рис. 4.3 показаны возможные коды интерфейсного состояния, выдаваемые БИФ, при выполнении команды УС. После установления функциональной связи с соответствующим БИФ с помощью команды УС БИФ остается подключенным к МВВ в течение всего времени до

информацией с БИФ и передачи в БИФ физических адресов устройств (ФАУ), с которыми осуществляется обмен микроЭВМ в текущем цикле. Шина КОМАНДЫ предназначена для передачи в БИФ кодов команд от процессора; СОСТОЯНИЕ — для сообщения процессору кодов состояния БИФ; ВЫЗОВ — для подачи процессором в БИФ сигнала ВЫЗОВ при выдаче любой команды; ОТВЕТ — для информирования процессора об окончании приема команды от процессора; ЗАПРОС — для выставления про-

Таблица 4.3

Интерфейсная команда	Разряд ШК				Допустимый код на шине состояния	Примечание
	4	3	2	1		
Установить связь (УС)	0	0	0	0	000 010 110	Выдаются физические адреса устройств
Начальная установка (НУ)	0	0	0	1	000 001 010	
Разрешить запрос на прерывание (РЗП)	0	0	1	0	000 010	Сигнал ЗАПРОС формируется после выполнения одной из команд обмена либо по инициативе УВВ
Выдать состояние (ВС)	0	1	0	0	000 001 010 011 111	
Принять первый байт (ППБ)	1	0	0	0	000 001 010 011 111	
Принять байт (ПБ)	1	0	0	1	000 001 010 011 111	
Принять байт последний (ПБП)	1	0	1	0	000 001 010 011 111	
Принять команду (ПК)	1	0	1	1	000 001 010	
Выдать первый байт (ВПБ)	1	1	0	0	000 001 010 011 111	
Выдать байт (ВБ)	1	1	0	1	000 001 010 011 111	

Интерфейсная команда	Разряд ШК				Допустимый код на шине состояния	Примечание
	4	3	2	1		
Выдать байт последний (ВБП)	1	1	1	0	000 001 010 011 111	
Выдать уточненное состояние (ВУС)	1	1	1	1	000 010	Команда выдается только после выставления состояния КВУС (см. табл. 4.4). Уточненное состояние выдается по информационным шинам

Таблица 4.4

Интерфейсное состояние	Разряд ШС			Примечание
	3	2	1	
Авария (А)	0	0	0	—
Команда выполняется (КВП)	0	0	1	—
Команда выполнена (КВ)	0	1	0	—
Команда выполнена со сбоем (КВС)	0	1	1	—
Команда выполнена, есть запрос на прерывание (КВЗП)	1	1	0	Только на команду УС
Команда выполнена, есть уточненное состояние (КВУС)	1	1	1	—

тех пор, пока не появится новая команда УС с другим ФАУ, которая и отключит его от МВВ. Таким образом, команда УС подготавливает БИФ с заданным ФАУ к последующему обмену информацией с процессором.

По команде ПРИНЯТЬ КОМАНДУ в БИФ по информационной магистрали передается код команды, которую необходимо выполнить. Алгоритм выполнения команды представлен на рис. 4.4.

Состояние «Команда выполняется» указывает на тот факт, что БИФ к данному моменту начал, но не закончил выполнение команды. Это состояние может также выдаваться БИФ после получения команды НАЧАЛЬНАЯ

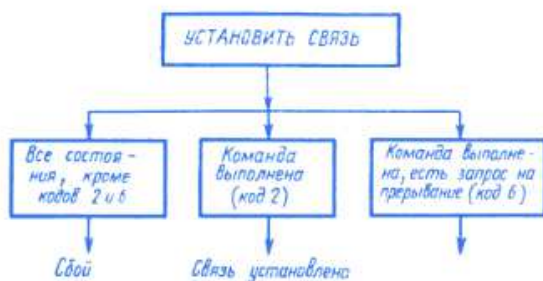


Рис. 4.3. Возможные коды интерфейсного состояния, выдаваемые БИФ при выполнении команды УС



Рис. 4.4. Алгоритм выполнения БИФ команды ПРИНЯТЬ КОМАНДУ



Рис. 4.5. Алгоритм выполнения БИФ команды ВЫДАТЬ УТОЧНЕННОЕ СОСТОЯНИЕ

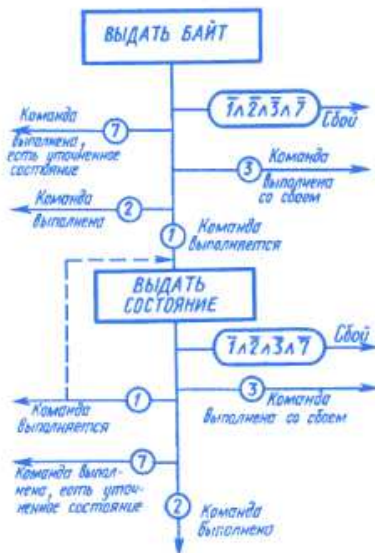


Рис. 4.6. Алгоритм выполнения БИФ команды ВЫДАТЬ БАЙТ (ВЫДАТЬ ПЕРВЫЙ БАЙТ, ВЫДАТЬ БАЙТ ПОСЛЕДНИЙ)

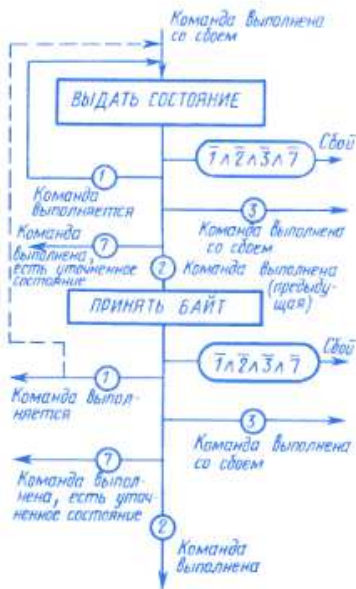


Рис. 4.7. Алгоритм выполнения БИФ команды ПРИНЯТЬ БАЙТ (ПРИНЯТЬ ПЕРВЫЙ БАЙТ, ПРИНЯТЬ БАЙТ ПОСЛЕДНИЙ)

Рис. 4.8. Алгоритм выполнения БИФ команды РАЗРЕШИТЬ ЗАПРОС НА ПРЕРЫВАНИЕ



УСТАНОВКА или **ВЫДАТЬ СОСТОЯНИЕ**. Состояние «Команда выполнена, есть уточненное состояние» устанавливается в том случае, если в результате выполнения команды БИФ или УВВ перешли в некоторый специфический режим, который отличен от режима, соответствующего состоянию «Команда выполнена». При этом процессор с помощью команды **ВЫДАТЬ УТОЧНЕННОЕ СОСТОЯНИЕ** получает от БИФ по информационным магистралям код его состояния. Алгоритм выполнения команды **ВЫДАТЬ УТОЧНЕННОЕ СОСТОЯНИЕ** представлен на рис. 4.5.

Алгоритм выполнения команды **ВЫДАТЬ БАЙТ** (**ВЫДАТЬ ПЕРВЫЙ БАЙТ**, **ВЫДАТЬ БАЙТ ПОСЛЕДНИЙ**) представлен на рис. 4.6. В процессе выполнения команды БИФ подготавливает и устанавливает на информационной магистрали байт информации. Если к моменту окончания цикла обмена БИФ не успевает установить передаваемый байт, то на шинах состояния магистрали выставляется интерфейсное состояние «Команда выполняется». Данное состояние может быть также выдано БИФ после получения команд **НАЧАЛЬНАЯ УСТАНОВКА**, **ВЫДАТЬ СОСТОЯНИЕ**, **ПРИНЯТЬ КОМАН-**

ДУ. Для завершения обмена информацией после получения кода состояния «Команда выполняется» процессор посылает БИФ команду **ВЫДАТЬ СОСТОЯНИЕ** до тех пор, пока БИФ не выдаст байт информации о своем состоянии на информационные магистрали.

Алгоритм выполнения команды **ПРИНЯТЬ БАЙТ (ПРИНЯТЬ ПЕРВЫЙ БАЙТ, ПРИНЯТЬ БАЙТ ПОСЛЕДНИЙ)** приведен на рис. 4.7.

Алгоритм выполнения команды **РАЗРЕШИТЬ ЗАПРОС НА ПЕРЕРЫВАНИЕ (РЗП)** представлен на рис. 4.8. Получив команду, БИФ запоминает факт ее подачи. Выдача сигнала **ЗАПРОС** производится БИФ после окончания выполнения предыдущей команды приема байта информации и по запросу от УВВ. Команда РЗП переводит БИФ из пассивного режима в активный, при этом запрос на прерывание формируется только при наличии соответствующего запроса от УВВ. При получении от процессора команды **УСТАНОВИТЬ СВЯЗЬ** БИФ выдает информацию по шинам состояния «Команда выполнена, есть запрос на прерывание». Сигнал **ЗАПРОС** снимает БИФ после снятия сигнала **ОТВЕТ** при завершении выполнения команды **УСТАНОВИТЬ СВЯЗЬ**. На следующих циклах БИФ, с которым установлена связь, готов к обмену информацией.

Если, находясь в активном режиме, БИФ-источник информации принял байт информации и выставил сигнал **ЗАПРОС**, то эта информация сохраняется до поступления в БИФ команд **ВЫДАТЬ ПЕРВЫЙ БАЙТ, ВЫДАТЬ БАЙТ, ВЫДАТЬ БАЙТ ПОСЛЕДНИЙ**.

Подача в БИФ, работающий в активном режиме, любой интерфейсной команды (кроме команд УС и РЗП) переводит его в пассивный режим, при этом сигнал **ЗАПРОС**, если он был выставлен, снимается.

Алгоритм выполнения команды **НАЧАЛЬНАЯ УСТАНОВКА** представлен на рис. 4.9. По этой команде сбрасываются все элементы памяти БИФ и управления УВВ; связь, установленная по команде УС, сохраняется.

На рис. 4.10, а приведены временные диаграммы процессов обмена информацией по интерфейсу процессора с БИФ-источником и БИФ-приемником [11]. При обмене с БИФ-приемником информация и команда должны быть установлены на информационной магистрали до появления сигнала **ВЫЗОВ** и не должны меняться в течение всего его действия (рис. 4.10, а). Начиная с момента появления сигнала **ВЫЗОВ** от процессора, но не позднее

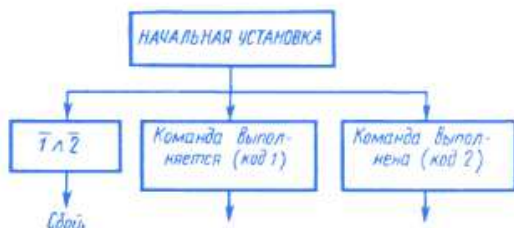


Рис. 4.9. Алгоритм выполнения БИФ команды НАЧАЛЬНАЯ УСТАНОВКА.

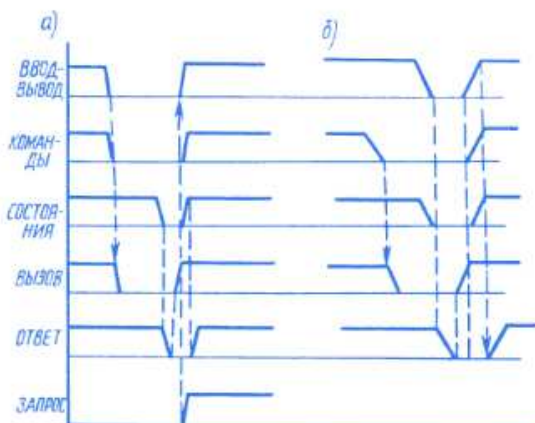


Рис. 4.10. Временные диаграммы процессоров обмена информацией процессора с БИФ-приемником (а) и БИФ-источником (б)

чем через 2 мкс, БИФ подготавливает информацию о степени выполнения команды, выставляя код состояния на соответствующих шинах. Формированием сигнала ОТВЕТ БИФ указывает процессору, что команда и информация восприняты. При получении этого сигнала процессор считывает код состояния БИФ на интервале времени от момента поступления сигнала ОТВЕТ до момента снятия сигнала ВЫЗОВ. Сигнал ОТВЕТ снимается БИФ после окончания сигнала ВЫЗОВ не позднее чем через 2 мкс, а код состояния снимается с шин состояния магистрали ввода — вывода не позднее чем через 200 нс.

При получении процессором информации от БИФ-источника процессор инициирует начало обмена путем

выдачи соответствующего кода команды на шину команд и подает БИФ сигнал ВЫЗОВ (рис. 4.10, б). Не более чем через 2 мкс БИФ устанавливает на информационной магистрали и шинах состояний информацию и посылает процессору сигнал ОТВЕТ, указывающий на завершение приема команды. Коды при этом не должны меняться на интервале времени от подачи сигнала ОТВЕТ до снятия процессором сигнала ВЫЗОВ. После снятия сигнала ВЫЗОВ сигнал ОТВЕТ снимается БИФ не более чем через 1 мкс. После снятия сигнала ВЫЗОВ код состояния БИФ должен быть снят не позднее чем через 200 нс. Необходимо помнить, что сигнал ВЫЗОВ выставляется БИФ только в том случае, если снят сигнал ОТВЕТ.

4.4. Системное программное обеспечение

Системное программное обеспечение включает в себя такие функциональные блоки, как супервизор, транслятор, интерпретатор, мониторы ввода — вывода, и решает следующие задачи в микроЭВМ:

- реализует входной язык;
- автоматически распределяет ресурсы машины между параллельно выполняемыми программами и устройствами ввода — вывода;
- управляет работой устройств ввода — вывода;
- обеспечивает диагностику микроЭВМ.

В зависимости от исполнения машины математическое обеспечение может быть записано в ПЗУ или размещено на другом носителе. В микроЭВМ может реализовываться либо монорежим, при котором выполняется одна задача, либо мультирежим, при котором возможно одновременное решение до четырех задач. При этом информация о выполняемых задачах хранится в оперативной памяти ПИД.

Основные функции супервизора в микроЭВМ заключаются в управлении процессом решения задач. При этом супервизор распределяет и защищает области памяти, отведенные для каждой из них. По сигналам прерываний от внешних устройств и выполняемых программ он автоматически управляет работой каналов ввода — вывода машины. Супервизор управляет порядком обслуживания задач, находящихся в кольцевой очереди, а также регистрирует прерывания, возникающие при этом. Прерывания могут возникать из-за:

— выполнения операторов входного языка, требующих прерывания обработки;

— ожидания завершения ввода — вывода информации;

— аварийного останова из-за переполнения памяти или сбоев в работе устройств ввода — вывода.

Транслятор переводит программу со входного языка в машинный и обеспечивает контроль синтаксической и семантической правильности вводимых программ и команд. Основным элементом трансляции является строка. Трансляция осуществляется в три этапа:

1) трансляция по очередной вводимой строке;

2) коррекция машинной программы вычислений после редактирующих воздействий;

3) формирование зоны значений переменных.

В результате трансляции описание каждой строки в машинных кодах подсоединяется к программе, формируя описание переменных.

В процессе редактирования отдельные строки могут исключаться и добавляться к программе. В памяти программа вычислений располагается в порядке поступления строк. Вычисления по программе производятся в порядке их нумерации и осуществляются интерпретатором. В процессе проведения расчетов он также перераспределяет свободную область памяти, подготавливает задания на ввод — вывод, а также информирует о всевозможных ошибках.

Мониторы ввода — вывода непосредственно управляют устройствами ввода — вывода в соответствии с заданием программы.

4.5. Входной язык

Управление работой машины осуществляется на базе алгоритмического языка БЭЙСИК. На микроЭВМ реализована одна из наиболее мощных версий языка БЭЙСИК, включающая в себя расширенные средства обработки текстов, матричные операторы, операторы сортировки и поиска данных в массивах.

Программист работает на микроЭВМ только в среде языка БЭЙСИК, дополненного набором команд для ввода, редактирования, отладки и запуска программ в диалоговом режиме. Файловая система, построенная на базе катализированных наборов данных, позволяет легко создавать, копировать, удалять отдельные файлы. В ма-

шине может быть реализована одна из версий языка БЭЙСИК (БЭЙСИК-1, БЭЙСИК-2). Основное отличие языка БЭЙСИК-2 от БЭЙСИК-1 — наличие расширенной и усовершенствованной таблицы кодов специальных клавиш клавиатуры, некоторых дополнительных операторов, а также расширенные функции и синтаксис некоторых операторов и команд.

Для управления интерпретирующей системой, а также ходом вычислительного процесса с целью ввода — вывода, корректировки текста программ, осуществления операций с памятью и внешними устройствами в микроЭВМ предусмотрены системные команды, вызываемые нажатием специальных клавиш на клавиатуре: CLEAR, CONTINUE*, HALT/STEP*, STMT NUMBER*, LIST, LOAD, SAVE, RESET*, RENUMBER, RUN, TRACE, CR/LF*. Команды, отмеченные знаком *, выполняются сразу же после нажатия соответствующей клавиши; остальные — после ввода их в ЭВМ и нажатия специальной клавиши CR/LF.

На клавиатуре имеется 16 клавиш специальных функций, используемых как в верхнем, так и в нижнем регистрах и таким образом определяющих 32 функции. Эти клавиши являются программно-управляемыми и программируемыми при выполнении текущей программы. Программирование клавиш осуществляется с помощью оператора DEFFN. Использование клавиш специальных функций позволяет:

- извлекать из памяти наиболее часто встречающиеся фрагменты текста;

- обращаться к фиксированным подпрограммам, по аналогии с выполнением оператора GOSUB в программе.

С подробным описанием операторов, команд, форматов переменных, типов данных и переменных языка БЭЙСИК-2 можно ознакомиться в [12].

4.6. Пакеты прикладных программ

Широкое использование микроЭВМ «Искра-226» в различных областях науки и техники привело к созданию большого числа пакетов прикладных программ. Кратко приведем назначение основных из них.

1. Система ТЕЛЕКОМ предназначается для написания и отладки загружаемого математического обеспечения телекоммуникационного интерфейса (БИФ 015-85 или БИФ 015-87) и специализированных драйверов ка-

нального процессора, а также для подготовки программных сегментов, содержащих внешнее математическое обеспечение и драйверы в виде, пригодном для использования в БЭЙСИК-программах [12]. Систему можно использовать для подготовки математического обеспечения любой микропроцессорной системы, а сегменты программ, не зависящие от конкретной архитектуры, можно отладить и протестировать при помощи отладочного монитора. Для работы системы достаточно минимальной конфигурации микроЭВМ «Искра-226» (исполнение 1). Программы системы в своей работе используют гибкий или жесткий диск.

2. Локальная информационно-поисковая система ПРИМА предназначена для создания и обслуживания персональных баз данных (БД) пользователей широкого профиля [13]. Взаимодействие пользователя с системой позволяет создавать БД, реорганизовывать их, осуществлять поиск информации в БД, а также представлять их в различных видах. Система применима в области обработки научно-технической информации. Для работы системы требуется микроЭВМ «Искра-226», накопитель на гибких дисках, устройство печати.

3. Комплекс САГРАФ — интегрированная система машинной геометрии и графики — предназначена для автоматизации различных геометрических и графических работ. Прикладное программное обеспечение используется в локальной информационно-поисковой системе ПРИМА, а также для автоматизации научных исследований и в автоматических обучающих системах. В комплект программ входят следующие компоненты [14]:

- обучающая система ТРЕНАЖЕР;
- прикладные программы построения графиков, прямоугольных и круговых диаграмм;
- прикладная программа графического отображения функций двух переменных;
- диалоговая система ГРАФИЧЕСКИЙ ПРОЦЕССОР, предназначенная для графической интерпретации результатов обработки в виде графиков, круговых и прямоугольных диаграмм при работе с информационными системами.

Для реализации комплекса необходима минимальная конфигурация микроЭВМ «Искра-226» (исполнение 1).

4. Интеллектуальный терминал ИТ/«Искра-226» предназначена для теледоступа к базам данных, для работы в сети передачи данных и в системе распределе-

ния времени. Пакет применяется для распределенной обработки научно-технической информации, обучения пользователя работе с интерактивными системами, а также в различных системах телеобработки данных [15].

Для работы системы необходима минимальная конфигурация микроЭВМ и телекоммуникационный интерфейс БИФ «Искра 015-85».

Широкое распространение микроЭВМ «Искра-226» в различных областях народного хозяйства приводит к появлению все новых пакетов прикладных программ, решающих задачи обработки и преобразования информации.

Приложение

Универсальный программный интерфейс



Универсальный программный интерфейс (УПИ) — набор системных вызовов, совместимых с любой операционной системой СМ 1810. Если некоторая программа использует системные вызовы УПИ и не использует системных вызовов, специфических для конкретной операционной системы, то эту программу можно перенести с одной операционной системы на другую. На рис. П.1 показаны соотношения между программой (кодом применения), аппаратными средствами и внутренними программными слоями. Направленные вниз стрелки представляют поток команд и поток данных от кода применения к аппаратным средствам, где команды в конечном счете должны быть выполнены. (На рисунке не приведены стрелки, показывающие направление потока данных от аппаратных средств к коду применения.) Перечеркнутая стрелка, направленная вниз, указывает на то, что код применения не использует прямых вызовов операционной системы. Вместо этого любой диалог между кодом применения и операционной системой осуществляется УПИ.

Для переноса применения из среды одной операционной системы в другую необходимо создать библиотеки УПИ для другой операционной системы. Эти библиотеки всегда осуществляют один и тот же интерфейс для применения, хотя их интерфейс с операционной системой сконфигурирован только для данной операционной системы.

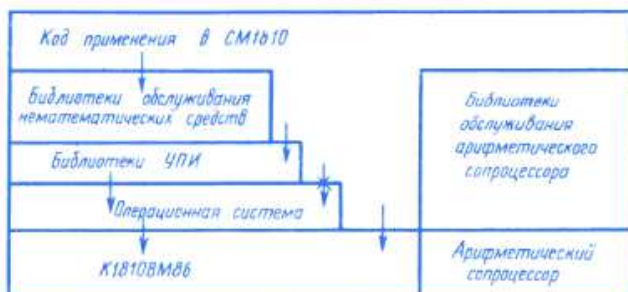


Рис. П.1. Соотношение от кода применения к аппаратным средствам

Библиотеки УПИ имеются для операционных систем ДОС 1810, БОС 1810 и МДОС 1810. Предполагается разработка библиотек УПИ для Микрос-86 и ОС СФП 1810.

П.1. Системные вызовы управления памятью

Программы для операционной системы БОС 1810 при загрузке и исполнении размещаются в заданном объеме памяти в зависимости от их конфигурации. Часть памяти, не занятая загруженным кодом программы или данными, — *пул свободного пространства* — доступен программам динамически, т. е. во время исполнения программ. Операционная система управляет памятью как *сегментами*, которые программы могут получать, использовать и возвращать. Программы используют системные вызовы DQ\$ALLOCATE и DQ\$FREE для получения сегментов памяти из пула и возврата сегментов в пул соответственно. Они могут использовать вызов DQ\$GET\$SIZE для получения информации о размещенных сегментах памяти.

П.2. Системные вызовы управления файлами

Почти половина системных вызовов УПИ используется для управления файлами. На рис. П.2 приводятся порядок и взаимосвязь наиболее часто используемых системных вызовов управления файлами.

Ключом к использованию файлов является *присоединение*. Чтобы присоединение к файлу и затем использовать его для выполнения действий. Одновременно другие программы могут иметь собственные присоединения к тому же файлу. Каждая программа, обладающая присоединением к файлу, использует его для исключительного доступа к файлу.

Программа получает присоединение при помощи вызова DQ\$ATTACH (если файл уже существует) или DQ\$CREATE (для создания нового файла). Если программа больше не нуждается в присоединении, то она может использовать вызов DQ\$DETACH для удаления присоединения. Для удаления как присоединения, так и файла программа использует вызов DQ\$DELETE.

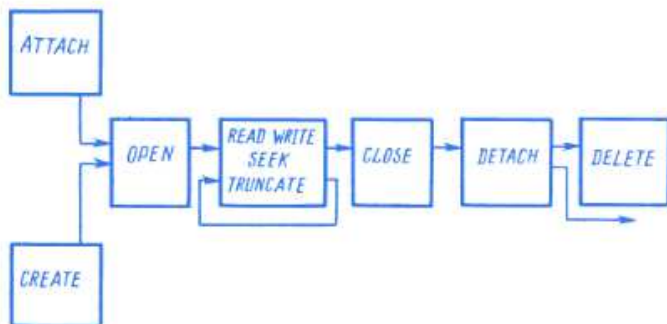


Рис. П.2. Взаимоотношение системных вызовов управления файлами

Если программа обладает присоединением, то она может вызвать `DQ$OPEN`, чтобы подготовить присоединение для операций ввода — вывода при помощи вызовов `DQ$READ` и `DQ$WRITE`. Она может переместить указатель файла, связанный с присоединением (вызов `DQ$SEEK`). Когда программа заканчивает ввод — вывод, она закрывает присоединение (вызов `DQ$CLOSE`). Заметим, что программа открывает и закрывает присоединение, а не файл. В том случае, когда программа не удаляет присоединение, она может продолжать открывать и закрывать его.

Если программа осуществляет вызов `DQ$DELETE` для удаления файла, то файл не может быть удален до тех пор, пока к нему существуют другие присоединения. Файл помечается для удаления, но в действительности не удаляется до тех пор, пока не будут удалены все присоединения. К такому файлу не может быть создано ни одно новое присоединение.

П.3. Коды возврата и системные вызовы обработки особых ситуаций

Каждый системный вызов УПИ, за исключением `DQ$EXIT`, возвращает значение кода возврата, определяющего результат исполнения. Любой код возврата имеет уникальное mnemonic имя. Например, код 0, указывающий на то, что не было никаких ошибок или необычных ситуаций, имеет имя `E$OK`. Другие коды возврата, означающие наличие особых ситуаций, классифицируются следующим образом:

- ошибки среды. Вызываются условиями вне управления программой; например, ошибки устройства или недостаточная память;

- ошибки программиста. Вызываются ошибками программирования (например, «плохой параметр»), хотя «деление на ноль», «переполнение», «проверка области» и ошибки, обнаруженные арифметическим сопроцессором, также классифицируются как ошибка программиста.

При обнаружении особой ситуации обычным (по умолчанию) действием системы является вывод сообщения об ошибке на консоль и прекращение программы. Однако программа может назначить собственный драйвер обработки особых ситуаций при помощи вызова `DQ$TRAP-EXCEPTION`. Этот драйвер при необходимости интерпретирует коды возврата при помощи вызова `DQ$DECODE$EXCEPTION`;

П.4. Вызовы УПИ из программ

В качестве примеров рассмотрим использование системного вызова `DQ$ALLOCATE` в программах, написанных на языках ПЛ/М-86 и макроассемблере. Эти примеры можно обобщить на другие системные вызовы УПИ.

Синтаксис системного вызова `DQ$ALLOCATE` записывается следующим образом:

```
base$addr = DQ$ALLOCATE (size, except$ptr);
```

Здесь имеются три параметра: `size` (тип данных `WORD`), `except$ptr` (тип данных `POINTER`) и `base$addr` (тип данных `WORD` или `SELECTOR` в зависимости от версии ПЛ/М-86).

В каждом из примеров запрашивается 128 байт памяти и указывается на слово с именем `ERR`, в которое должен быть помещен код возврата.

Пример вызывающей последовательности на языке ПЛ/М-86:

```
DECLARE    ARRAY    BASE        WORD; (или SELECTOR)
           ERR
ARRAYBASE = DQ$ALLOCATE    (128, @ ERR);
```

Пример вызывающей последовательности на языке макроассемблера ASM86:

```
MOV AX, 128
PUSH AX                ; первый параметр
LEA AX, ERR
PUSH AX                ; второй параметр
CALL DQ$ALLOCATE
MOV ARRAYBASE, AX     ; возвращаемая величина
```

П.5. Описание системных вызовов

Рассмотрим все системные вызовы универсального программного интерфейса.

При этом примем следующие обозначения типов данных языка ПЛ/М-86:

<i>байт</i> — BYTE;	<i>селектор</i> — SELECTOR;
<i>слово</i> — WORD;	<i>обозначение</i> — TOKEN;
<i>указатель</i> — POINTER	<i>строка</i> — STRING.

TOKEN и STRING — специфические типы данных, которые обычно не используются при написании программ, не основанных на УПИ.

DQ\$ALLOCATE запрашивает сегмент памяти из пула свободной памяти

`base$addr = DQ$ALLOCATE (size, except$ptr);`

Входной параметр

size — *слово*, которое указывает размер:

— если не нуль, то содержит размер запрошенного сегмента в байтах. Если параметр размера не кратен 16, то он округляется до ближайшего целого, кратного 16, перед тем как запрос на размещение будет обрабатываться;

— если нуль, то указывает, что размер запроса равен 65 536 (64 К) байт.

Выходные параметры:

base\$addr — *селектор*, в который операционная система помещает базовый адрес сегмента памяти. Если запрос неудачен из-за того, что запрошенная память не может быть выделена, то будет равен 0FFFFH и система вернет код особой ситуации E\$MEM; .

except\$ptr — *указатель* на *слово*, в которое система помещает код возврата.

О п и с а н и е. Этот вызов используется для запроса дополнительной памяти из пула свободной памяти программы.

DQ\$ATTACH создает присоединение к существующему файлу:

`connection = DQ$ATTACH (path$ptr, except$ptr);`

Входной параметр

`path$ptr` — указатель на строку, содержащую полное имя файла, который должен быть присоединен.

Выходные параметры:

`connection` — *обозначение* для присоединения файла;

`except$ptr` — *указатель на слово*, в которое система помещает код возврата.

О п и с а н и е. Этот вызов позволяет программе получить присоединение к любому файлу. Допускается присоединение файла, который уже присоединен (ранее выполненные присоединения остаются назначенными).

Программа может использовать вызов `DQ$RESERVE$IO$MEMORY` для резервирования памяти, необходимой УПИ для внутренних структур данных при вызове программой `DQ$ATTACH` и для буферов при вызове программой `DQ$OPEN`. Преимущество резервирования памяти заключается в гарантии того, что память будет доступна, когда это необходимо. Если память не зарезервирована, то вызов `DQ$ATTACH` может быть неудачным из-за недостатка памяти.

`DQ$CHANGE$ACCESS` позволяет изменить права доступа владельца файла или пользователя с идентификатором `WORLD` (общий):

`CALL DQ$CHANGE$ACCESS (path$ptr, user, access, except$ptr);`

Входные параметры:

`path$ptr` — *указатель на строку*, содержащую полное имя файла;
`user` — *байт*, определяющий тип пользователя, доступ которого должен быть изменен:

Величина	Смысл
0	Владелец файла
1	WORLD (все пользователи в системе)

`access` — *байт*, определяющий тип доступа, который гарантируется пользователю. Биты в данном байте означают следующее (бит 0 является младшим битом):

Бит	Смысл
0	Пользователь может уничтожить файл или директорию
1	Чтение (файла) или вывод содержимого (директории)
2	Добавление (файла) или новых элементов (директорий)
3	Изменение (чтение и запись файла) или изменение содержимого (директории)
4—7	Должен быть нулем

Выходной параметр

`except$ptr` — *указатель на слово*, в которое система помещает код возврата.

Описание. В среде БОС1810 любая программа связана с объектом-пользователем, обычно называемым *пользователем по умолчанию*. Пользователь по умолчанию состоит из одного или нескольких идентификаторов пользователей. Каждый файл имеет связанный с ним набор пар «идентификатор пользователя — маска доступа», при этом маска определяет права доступа, которые пользователь с соответствующим идентификатором имеет к данному файлу. Если программа вызывает `DQ$CREATE` для создания файла или `DQ$ATTACH` для получения другого присоединения к файлу, то результирующее присоединение будет иметь все права доступа, соответствующие идентификаторам пользователей, относящиеся как к файлу, так и к пользователю по умолчанию. Системный вызов `DQ$CHANGE$ACCESS` изменяет для конкретного файла права доступа, связанные с конкретным идентификатором пользователя.

В среде БОС1810 пользователь по умолчанию имеет два идентификатора (ID): ID владельца, связанный с программой; `WORLD`, связанный со всеми программами. Вызов `DQ$CHANGE$ACCESS` может изменить для файла права доступа как владельца, так и `WORLD`.

`DQ$CHANGE$EXTENSION` позволяет изменить или добавить расширение к концу хранящегося в памяти имени файла. Он не воздействует на имя файла на томе внешней памяти:

```
CALL DQ$CHANGE$EXTENSION (path$ptr, extension$ptr;  
    except$ptr);
```

Входные параметры:

`path$ptr` — *указатель на строку*, определяющую полное имя файла, который должен быть переименован;

`extension$ptr` — *указатель на серию из 3 байт*, содержащих знаки, которые добавляются к полному имени. Это не строка. Необходимо включить 3 байт, даже если некоторые из них пусты.

Выходной параметр

`except$ptr` — *указатель на слово*, в которое система помещает код возврата.

Описание. Этот вызов является средством для редактирования строк, хранящихся в памяти и представляющих имя файла. Если существующее имя файла имеет расширение, то вызов `DQ$CHANGE$EXTENSION` заменяет данное расширение указанными тремя знаками. В противном случае вызов `DQ$CHANGE$EXTENSION` добавляет эти три знака в качестве расширения имени файла.

Например, компилятор может использовать вызов `DQ$CHANGE$EXTENSION` для редактирования имени исходного файла, такого, как `:AFDI:FILE.SRC`, в имя объектного файла, такое, как `:AFDI:FILE.OBJ`, а затем создать объектный файл.

`DQ$CLOSE` ожидает завершения операций с файлом (если такие есть), опустошает выходные буфера и освобождает любые буфера, связанные с присоединением:

```
CALL DQ$CLOSE (connection, except$ptr);
```


Входной параметр

`connection` — *обозначение* для присоединения файла, которое открыто в текущий момент.

Выходной параметр

`except$ptr` — *указатель на слово*, в которое система помещает код возврата.

Описание. Этот системный вызов закрывает присоединение, которое было открыто системным вызовом `DQ$OPEN`. Он выполняет следующие шаги:

- 1 — ожидает окончания операций ввода — вывода;
- 2 — вызывает запись в файл любой информации в частично заполненном буфере;
- 3 — освобождает любые связанные с файлом буфера;
- 4 — закрывает присоединение к файлу. При этом присоединение является действительным и может быть повторно открыто.

`DQ$CREATE` создает новый файл и устанавливает присоединение к файлу:

```
connection = DQ$CREATE (path$ptr, except$ptr);
```

Входной параметр

`path$ptr` — *указатель на строку*, задающую полное имя файла, который должен быть создан.

Выходные параметры:

`connection` — *обозначение* для присоединения к файлу;

`except$ptr` — *указатель на слово*, в которое система помещает код возврата.

Описание. Этот системный вызов создает файл с заданным именем и возвращает присоединение в программу. Если файл с таким именем существует, то выполняется его «усечение» до нулевой длины и данные в нем разрушаются. Чтобы предотвратить случайное разрушение файла, следует выдать вызов `DQ$ATTACH` перед выдачей вызова `DQ$CREATE`. Если файл не существует, то при обращении к вызову `DQ$ATTACH` будет получен код особой ситуации `E$FNEXIST`.

`DQ$DECODE$EXCEPTION` переводит код особой ситуации в соответствующую *мнемонику*:

```
CALL DQ$DECODE$EXCEPTION (except$code, buff$ptr, except$ptr);
```

Входной параметр

`except$code` — *слово*, содержащее значение кода особой ситуации, которое должно быть интерпретировано.

Выходные параметры:

`buff$ptr` — *указатель на буфер* (содержащий не менее 81 байт), в который система вернет *строку*;

`except$ptr` — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Программа может осуществить вызов `DQ$DECODE$EXCEPTION` для изменения значения кода особой ситуации на его шестнадцатеричный эквивалент, сопровождаемый мнемоникой. Например, если в вызове `DQ$DECODE$EXCEPTION` указано, что `except$code = 2`, то система вернет в область, указанную параметром `buff$ptr`, строку

```
002: E$MEM
```

DQ\$DECODE\$TIME возвращает текущие системные время и дату как целое двойное слово и как серию байтов, содержащих знаки:

```
CALL DQ$DECODE$TIME (time$ptr; except$ptr);
```

Выходные параметры:

`time$ptr` — указатель на структуру следующей формы:

```
DECLARE DT STRUCTURE(  
    SYSTEM$TIME DWORD,  
    DATE(8) BYTE,  
    TIME(8) BYTE);
```

Если значение в элементе `SYSTEM$TIME` равно 0 при вызове `DQ$DECODE$TIME`, то возвращаются текущие дата и время, как это описано ниже (см. описание для информации о формате).

Если значение в `SYSTEM$TIME` не равно 0 при вызове `DQ$DECODE$TIME`, то предполагается, что значением `SYSTEM$TIME` является число секунд с полуночи 1 января 1978 г., это значение декодируется и возвращается в полях `DATE` и `TIME`.

`except$ptr` — указатель на слово, в которое система помещает код возврата.

Описание. Этот системный вызов возвращает текущие дату и время как серию байтов (заметим, что это не строка).

`ДАТА` имеет форму `MM/DD/YY` (месяц, день и год). Две косые черты (/) располагаются в третьем и шестом байтах. Например, `ДАТА` 15 января 1982 г. возвращается как

```
01/15/82
```

`ВРЕМЯ` имеет форму `HH:MM:SS` (часы, минуты и секунды) с разделительными двоеточиями (:). Значение `HH` лежит в интервале от 0 до 23. Например, `ВРЕМЯ` 3 часа 12 минут 20 секунд после полудня возвращается как

```
15:12:20
```

Если при вызове `DQ$DECODE$TIME` элемент `SYSTEM$TIME` равен нулю, то вызов сначала установит системное время (число секунд с 1 января 1978 г.) и затем декодирует его в описанную серию байтов.

Если элемент `SYSTEM$TIME` не нуль, то вызов `DQ$DECODE$TIME` использует его значение в качестве времени, которое должно быть декодировано.

Программа может сначала использовать вызов `DQ$FILE$INFO` для установки двух значений *двойных слов*, связанных с файлом (последнее время изменения файлов и время создания файла), а затем с помощью вызова `DQ$DECODE$TIME` интерпретировать время.

DQ\$DELETE удаляет существующий файл;

```
CALL DQ$DELETE (path$ptr, except$ptr);
```

Входной параметр

`path$ptr` — *указатель на строку*, содержащую полное имя файла, который должен быть удален

Выходной параметр

`except$ptr` — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Программа может использовать этот системный вызов для удаления файла. Непосредственным действием его является маркировка файла для удаления, а не собственно удаление, так как файл нельзя удалить до тех пор, пока существуют присоединения к нему. Вызов `DQ$DELETE` удалит файл только тогда, когда все существующие присоединения к файлу будут отсоединены. С того момента, как файл маркирован, никакие новые присоединения к нему не могут быть получены при помощи вызова `DQ$ATTACH`.

`DQ$DETACH` удаляет присоединенные (но не файл), установленные вызовами `DQ$ATTACH` или `DQ$CREATE`:

`CALL DQ$DETACH (connection, except$ptr);`

Входной параметр

`connection` — *обозначение* для присоединения файла, которое должно быть удалено.

Выходной параметр

`except$ptr` — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Данный системный вызов удаляет присоединение файла. Если присоединение открыто, то системный вызов `DQ$DETACH` до удаления присоединения автоматически закроет его (см. `DQ$CLOSE`). Вызов `DQ$DETACH` также удалит файл, если он был маркирован для удаления, и это присоединение было последним.

`DQ$EXIT` передает управление из программы операционной системе. Он не возвращает ни одного значения вызывающей программе (даже кода возврата):

`CALL DQ$EXIT (end$code);`

Входной параметр

`end$code` — *слово*, содержащее причину прекращения программы.

Описание. Этот вызов заканчивает программу. Перед действительным окончанием все присоединения программы закрываются и отсоединяются, а память, распределенная под программу при помощи `DQ$ALLOCATE`, возвращается в пул.

`DQ$FILE$INFO` возвращает информацию о файле (дальнейшее описание — для случая BOC1810):

`CALL DQ$FILE$INFO (connection, mode, file$info$ptr, except$ptr);`

Входные параметры:

`connection` — обозначение присоединения к файлу;
`mode` — *байт*, указывающий, следует ли вызову `DQ$FILE$INFO` возвращать идентификатор владельца файла. Устанавливается следующим образом:

Значение	Смысл
0	Не возвращать ID владельца
1	Возвращать ID владельца

Выходные параметры:

`except$ptr` — *указатель* на слово, в которое операционная система помещает код возврата;

`file$info$ptr` — *указатель* буфера, в котором возвращается запрошенная информация. Эта структура имеет следующую форму:

```

DECLARE          FDATA          STRUCTURE(
  OWNER (15)     STRING,
  LENGTH        DWORD,
  TYPE          BYTE,
  OWNER$ACCESS  BYTE,
  WORLD$ACCESS  BYTE,
  CREATE$TIME   DWORD,
  LAST$MOD$TIME DWOTD,
  RESERVED (20) BYTE);
  
```

где

`OWNER` — *строка*, содержащая (если запрашивается) идентификатор владельца файла;

`LENGTH` — *длина* файла;

`TYPE` — *число*, указывающее тип файла следующим образом:

Значение	Тип файла
0	Файл данных
1	Директория

`OWNER $ACCESS` — *байт*, указывающий тип доступа, гарантированный владельцу. Если бит взведен, то это означает, что соответствующий тип доступа гарантируется; в противном случае данный тип доступа запрещен (бит 0 является младшим битом):

Бит	Тип доступа
0	Удаление
1	Чтение (файл данных) или вывод содержимого (директория)
2	Добавление (файл данных) или добавление элементов (директория)
3	Изменение (чтение и запись: файл данных) или изменение доступа (директория)

WORLD\$ACCESS — бит, определяющий тип доступа, гарантированный пользователю с ID = WORLD (всем пользователям в системе). Если бит взведен, то это означает, что соответствующий тип доступа гарантируется; в противном случае данный типа доступа запрещен (бит 0 является младшим битом):

Бит	Тип доступа
0	Удаление
1	Чтение (файл данных) или вывод содержимого (директория)
2	Запись (файл данных) или добавление элементов (директория)
3	Изменение (чтение и запись: файл данных) или изменение доступа (директория)

CREATE\$TIME — дата и время создания файла, выраженные как число секунд с 1 января 1978 г. (для их преобразования в знаки используется системный вызов **DQ\$DECODE\$TIME**);

LAST\$MOD\$TIME — дата и время последней модификации файла или директории. Для файлов данных под их модификацией понимается запись в файл или его усечение; для директорий означает, что их элементы были изменены или добавлены (для их преобразования в знаки используется системный вызов **DQ\$DECODE\$TIME**).

О п и с а н и е. Рассматриваемый вызов позволяет программам получить информацию о файле данных или директории.

DQ\$FREE возвращает операционной системе сегмент памяти, полученный ранее с помощью вызова **DQ\$ALLOCATE**:

CALL DQ\$FREE (base\$addr, except\$ptr);

Входной параметр

base\$addr — обозначение базового адреса сегмента, который должен быть удален. Данное значение является обозначением, полученным при использовании вызова **DQ\$ALLOCATE** для запроса сегмента.

Выходной параметр

except\$ptr — указатель на слово, в которое операционная система помещает код возврата.

О п и с а н и е. Данный вызов возвращает указанный сегмент в пул памяти, из которого он был получен вызовом **DQ\$ALLOCATE**.

DQ\$GET\$ARGUMENT возвращает аргументы из командной строки, введенной с системной консоли. Данная командная строка является либо

той, которая запрашивалась от оператора программой, содержащей вызов `DQGETARGUMENT`, либо командной строкой, введенной при написании программы:

```
delimit$char = DQ$GET$ARGUMENT (argument$ptr, except$ptr);
```

Входной параметр

`argument$ptr` — *указатель* на буфер, в котором будет получен аргумент в форме строки. Минимальная длина буфера должна быть не меньше 81 байт.

Выходные параметры:

`delimit$char` — *байт*, содержащий знак-разделитель;

`except$ptr` — *указатель* на слово, в которое операционная система помещает код возврата.

Описание. Программа может использовать вызов `GET$ARGUMENT` для получения аргументов из командной строки. Каждый вызов возвращает один аргумент и знак разделителя, сопровождающий аргумент.

Этот системный вызов применяется в двух случаях:

1) получение аргументов из командной строки, используемой для вызова программы с консоли. При этом командная строка находится в буфере, который автоматически предоставляется для этой цели;

2) получение аргументов из строк, введенных в ответ на запрос из программы. В этом случае программа должна использовать один и тот же буфер при вызовах `DQ$READ` и `DQ$GET$ARGUMENT`, что может быть обеспечено путем вызова `DQ$SWITCH$BUFFER` перед вызовом `DQGETARGUMENT`.

Разделитель возвращается только в том случае, если код возврата равен нулю. Операционной системой БЭС 1810 распознаются следующие разделители:

```
.) (= # ! $ % \ + - > < ~
```

а также пробел и любой другой знак с шестнадцатеричным значением от 0 до 20 H.

Перед возвратом аргумента система осуществляет редактирование содержимого командного буфера:

— выбрасывает из командной строки амперсанд (&) и точку с запятой (;);

— замещает несколько смежных пробелов, разделяющих два аргумента, на один пробел (табуляция интерпретируется как пробел);

— преобразует строчные знаки в прописные, если они не являются частью заключенной в кавычки строки.

При возврате аргументов система рассматривает строки, заключенные внутри парных одиночных или двойных кавычек так, как они введены, т. е. без редактирования. Заключающие кавычки не возвращаются как часть аргумента.

Пример. Проиллюстрируем аргументы и разделители, возвращаемые последовательными вызовами `DQGETARGUMENT`. Предположим, что содержимым буфера является текст

```
PLM86 LINKER. PLM PRINT (:LP:) NOLIST
```

Приведем возвращаемые аргументы для пяти вызовов `DQGETARGUMENT`:

Номер вызова	Возвращаемый аргумент	Возвращаемый разделитель
1	(05H)PLM86	Пробел
2	(0AH))LINKER.PLM	Пробел
3	(05H)PRINT	(
4	(04H):LP:)
5	(06H)NOLIST	cr

Аргумент возвращается в форме *строки*, где первый байт задает ее длину. Во втором вызове, где аргумент содержит 10 знаков, первый байт имеет значение 0AH. Последним разделителем для каждого вызова является возврат каретки (cr). При помощи этого разделителя программа определяет, что аргументов в командной строке больше нет.

DQ\$GET\$CONNECTION\$STATUS возвращает информацию о присоединении файла;

CALL DQ\$GET\$CONNECTION\$STATUS (connection, info\$ptr, except\$ptr);

Входной параметр

connection — *слово*, содержащее обозначение для присоединения, статус которого необходимо получить.

Выходные параметры:

except\$ptr — *указатель* на *слово*, в которое операционная система помещает код возврата;

info\$ptr — *указатель* на структуру, в которой операционная система помещает информацию о статусе. Структура имеет следующий формат:

```
DECLARE INFO STRUCTURE (
    OPEN           BYTE,
    ACCESS         BYTE,
    SEEK           BYTE,
    FILE$PTR      DWORD);
```

где

OPEN равен 1, если присоединение открыто, в противном случае равен 2;

ACCESS — привилегии доступа присоединения. Права доступа гарантируются при установке соответствующего бита в I (бит 0 является младшим битом);

Бит	Доступ
0	Удаление
1	Чтение
2	Запись
3	Изменение (чтение или запись)

SEEK — типы поддерживаемого позиционирования.

Значение	Смысл
0	Позиционирование не разрешается
3	Разрешается позиционирование вперед и назад

FILE\$PTR — целое *двойное слово*, указывающее текущую позицию в файле. Позиция определяется как число байтов с начала файла; первым является нулевой байт. После **FILE\$PTR** не определено, если файл не открыт или если позиционирование не поддерживается устройством. Например, операции позиционирования невозможны для построено-печатающего устройства.

Описание. Этот вызов используется для получения информации о присоединении файла. Например, можно использовать этот системный вызов. Если программа выполнила ряд операций чтения или записи, то для определения того, где находится указатель файла, необходимо использовать вызовы **DQ\$GET\$CONNECTION\$STATUS**,

DQ\$GET\$EXCEPTION\$HANDLER возвращает адрес текущего драйвера обработки особых ситуаций:

CALL DQ\$GET\$EXCEPTION\$HANDLER (address\$ptr, except\$ptr):

Выходные параметры:

address\$ptr — *указатель на указатель*, в котором операционная система возвращает точку входа для текущего драйвера обработки особых ситуаций;

except\$ptr — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Данный системный вызов возвращает программе адрес текущего драйвера обработки особых ситуаций. Этот адрес либо был задан в последнем вызове **DQ\$TRAP\$EXCEPTION**, либо является адресом системного драйвера обработки особых ситуаций по умолчанию.

Вызов **DQ\$GET\$EXCEPTION\$HANDLER** используется совместно с вызовами **DQ\$TRAP\$EXCEPTION** и **DQ\$DECODE\$EXCEPTION**.

DQ\$GET\$SIZE возвращает размер ранее размещенного сегмента памяти:

size = DQ\$GET\$SIZE (base\$addr, except\$ptr):

Входной параметр

base\$addr — *обозначение* базового адреса сегмента памяти, который был размещен системным вызовом **DQ\$ALLOCATE** (равен адресу, полученному вызовом **DQ\$ALLOCATE** при размещении сегмента).

Выходные параметры:

size — *слов*, которое операционная система устанавливает следующим образом:

— если не нуль, то содержит размер сегмента в байтах;

— если нуль, то указывает, что размер сегмента равен 65 536 (64 К) байт;

except\$ptr — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Рассматриваемый вызов возвращает размер сегмента в байтах. Размер сегмента может не быть равным размеру, запрошенному в вызове `DQ$ALLOCATE`, так как вызов `DQ$ALLOCATE` распределяет память 16-байтовыми параграфами. При запросе сегмента, размер которого не кратен 16, вызов `DQ$ALLOCATE` увеличит размер до ближайшей 16-байтовой границы перед обработкой запроса.

`DQGETSYSTEM$ID` возвращает строку, которая идентифицирует операционную систему, обеспечивающую среду для УПИ:

`CALL DQGETSYSTEM$ID (id$ptr, except$ptr);`

Выходные параметры:

`id$ptr` — указатель на 21-байтовый буфер, в который операционная система поместит строку, идентифицирующую операционную систему;
`except$ptr` — указатель на слово, в которое операционная система помещает код возврата.

Описание. Данный системный вызов в случае БОС 1810 возвращает строку

`BOS 1810`

сопровожаемую 13 пустыми знаками.

`DQ$OPEN` открывает файл для операций ввода — вывода, определяет тип доступа к файлу и задает число буферов, необходимых для поддержки операций ввода — вывода:

`CALL DQ$OPEN (connection, access, num$buf, except$ptr);`

Входные параметры:

`connection` — обозначение для присоединения файла, которое должно быть открыто;

`access` — байт, задающий способ использования присоединения для доступа к файлу:

Значение	Смысл
1	Только чтение
2	Только запись
3	Изменение (как чтение, так и запись)

`num$buf` — байт, содержащий число буферов, необходимых для присоединения. При задании значения, большего 0, косвенно запрашивается автоматическое выполнение «двойной буферизации», что означает чтение вперед и (или) запись назад.

Выходной параметр

`except$ptr` — указатель на слово, в которое операционная система помещает код возврата.

Описание. Этот системный вызов подготавливает присоединение для вызовов `DQ$READ`, `DQ$WRITE`, `DQ$SEEK` и `DQ$TRUNCATE`. Любое число присоединений к одному и тому же файлу может быть

открыто одновременно. Системный вызов `DQ$OPEN` выполняет следующее:

1. Создает запрошенные буферы.
2. Устанавливает указатель присоединения файла в нуль, т. е. указатель местоположения показывает, где в файле начать следующую операцию ввода — вывода.
3. Начинает читать вперед, если параметр `num$buf` больше нуля и параметр `access` равен 1 или 3.

`DQ$OVERLAY` вызывается корневым модулем для загрузки оверлейного модуля, если в системе используются оверлеи:

```
CALL DQ$OVERLAY (name$ptr, except$ptr);
```

Входной параметр

`name$ptr` — *указатель* из строку, которая содержит имя оверлейного модуля. Имя должно быть задано в прописных буквах

Выходной параметр

`except$ptr` — *указатель* на слово, в которое операционная система помещает код возврата.

Описание. Данный системный вызов вызывается корневым модулем в оверлейной системе в тех случаях, когда требуется загрузить оверлейный модуль. Необходимо обеспечить компоновку всей библиотеки УПИ с корневой частью программы, а не с одним из оверлеев. При этом следует использовать опцию `INCLUDE` для включения файла внешних имен УПИ при ассемблировании или компиляции корневой части программы. Включением файла в корневой модуль обеспечиваются внешние ссылки ко всем подпрограммам УПИ из корневого модуля. Это также предотвращает от наличия неразрешенных внешних ссылок при компоновке корневого модуля с оверлеями.

`DQ$READ` копирует байты из файла в буфер:

```
bytes$read=DQ$READ (connection, buff$ptr, bytes$max, except$ptr);
```

Входные параметры

`connection` — *обозначение* для присоединения к файлу. Это присоединение должно быть открыто для чтения или для чтения и записи, и указатель файла должен соответствовать первому читаемому байту;

`buff$ptr` — *указатель* на буфер, в который будут помещаться данные из файла;

`bytes$max` — *слово*, содержащее максимальное число байтов, которые надо прочитать из файла.

Выходные параметры

`bytes$read` — *слово*, содержащее действительное число прочитанных байтов. Это число всегда равно или меньше `bytes$max`;

`except$ptr` — *указатель* на слово, в которое операционная система помещает код возврата.

Описание. Данный системный вызов читает набор последовательных байтов из файла, связанного с присоединением. Набор байтов помещается в буфер, заданный вызывающей программой.

Параметр `buff$ptr` указывает операционной системе место, в которое следует поместить байты после их чтения. При этом программа предоставляет буфер, вызов `DQ$READ` копирует указанное число байтов

(если только он не обнаружит конец файла). Таким образом, если буфер имеет недостаточную длину, то операционная система запишет в область памяти за буфером.

Программа запрашивает максимальное число байтов, которое операционная система может поместить в буфер. Однако бывают обстоятельства, при которых операционная система читает меньшее число байтов:

— если вызов `DQ$READ` находит конец файла до завершения чтения запрошенного числа байтов, то этот системный вызов возвращает только предшествующие концу файла байты.

Параметр `bytes$read` может быть меньше параметра `bytes$max` и никаких особых ситуаций не возникает;

— если во время операции чтения произошла особая ситуация, то информация в буфере и значение параметра `bytes$read` являются незначимыми и должны игнорироваться.

Присоединение должно быть открыто для чтения или для чтения и записи. Если это не так, то вызов `DQ$READ` возвращает код особой ситуации.

DQ\$RENAME изменяет полное имя файла:

```
CALL DQ$RENAME (path$ptr, new$ptr, except$ptr);
```

Входные параметры:

`path$ptr` — указатель на строку, задающую полное имя файла, который должен быть переименован;

`new$ptr` — указатель на строку, задающую новое полное имя файла. Это имя не может быть именем существующего файла.

Выходной параметр

`except$ptr` — указатель на слово, в которое операционная система помещает код возврата.

Описание. Данный системный вызов позволяет программам изменять полные имена файлов или директорий. Следует отметить, что при переименовании директории изменяются полные имена всех содержащихся в ней файлов. При переименовании файла существующие к нему присоединения остаются назначенными.

Полное имя файла может быть изменено любым способом, при котором файл или директория остаются на одном и том же томе.

DQ\$RESERVE\$IO\$MEMORY позволяет резервировать такой объем памяти, который обеспечивает открытие и присоединение файлов, используемых программой:

```
CALL DQ$RESERVE$IO$MEMORY (number$files, number$buffers, except$ptr);
```

Входные параметры:

`number$files` — максимальное число файлов, которые программа может одновременно присоединить. Это значение не должно быть больше 12. Кроме того, не более шести файлов могут быть открыты одновременно;

`number$buffers` — общее число буферов (максимум 12), необходимых одновременно. Например, если программа имеет два одновременно открытых файла, для каждого из которых используется по два бу-

фера (заданы при открытии), то значение параметра `number$files` должно быть равно двум, а значение параметра `number$buffers` должно быть равно четырем.

Выходной параметр

`excert$ptr` — указатель на слово, в которое операционная система помещает код возврата.

Описание. Рассматриваемый вызов устанавливает память вне вызывающей программы, что гарантирует ее доступность для присоединения и открытия файлов. Эта память используется для внутренних структур данных УПИ при запросе программой присоединения к файлу с помощью вызова `DQ$ATTACH` и для буферов при открытии программой присоединения к файлу вызовом `DQ$OPEN`. Память, резервируемая при этом, не может быть размещена вызовом `DQ$ALLOCATE`. Программа должна осуществить вызов `DQ$RESERVE$IO$MEMORY` перед любым вызовом `DQ$ALLOCATE`.

В вызове `DQ$RESERVE$IO$MEMORY` можно задать до 12 файлов (которые могут быть присоединены при использовании резервируемой памяти) и до 12 буферов (которые могут быть запрошены при открытии файлов).

`DQ$SEEK` изменяет указатель файла, связанный с заданным присоединением:

`CALL DQ$SEEK (connection, mode, move$count, excert$ptr);`

Входные параметры:

`connection` — обозначение для открытого присоединения, указатель файла которого должен быть перемещен.

`mode` — *байт*, определяющий тип перемещения указателя файла, задаваемый следующим образом:

Режим	Смысл
1	Перемещает указатель назад на заданное в параметре <code>move\$count</code> число байтов. Если это число больше числа байтов до начала файла, то указатель устанавливается на первый байт (позиция 0).
2	Устанавливает указатель в позицию, заданную в параметре <code>move\$count</code> . Позиция 0 — первая в файле. Допустимо перемещение указателя за конец файла.
3	Перемещает указатель файла вперед на заданное в параметре <code>move\$count</code> число байтов. Допустимо перемещение указателя за конец файла.
4	Сначала перемещает указатель в конец файла, а затем назад на заданное в параметре <code>move\$count</code> число байтов. Если заданное перемещение приведет к перемещению за начало файла, то указатель устанавливается на первый байт в файле (позиция 0).

`move$count` — целое *двойное слово*, задающее в байтах перемещение указателя файла.

Выходной параметр

`exceptptr` — указатель на слово, в которое операционная система помещает код возврата.

Описание. При выполнении ввода — вывода, отличающегося от последовательного, в программах следует выполнить вызов для позиционирования указателя файла перед вызовами `DQ$READ`, `DQ$TRUNCATE` или `DQ$WRITE`. Позиция указателя файла определяет в файле место, откуда начнут выполняться вызовы `DQ$READ`, `DQ$WRITE` и `DQ$TRUNCATE`. Если программа выполняет последовательный ввод — вывод, то нет необходимости использовать данный системный вызов.

Возможно позиционирование указателя файла за конец файла. Если программа делает это и затем обращается к системному вызову `DQ$READ`, то операция чтения начинается с конца файла. Если программа вызывает `DQ$WRITE`, когда указатель файла находится за концом файла, то данные будут записаны за концом файла.

`DQ$SPECIAL` определяет, будет ли построчное редактирование возможно при вводе оператором информации с консоли:

`CALL DQ$SPECIAL (mode, connptr, exceptptr);`

Входные параметры:

`mode` — байт, используемый для указания режима ввода:

Значение	Смысл
1	Прозрачный
2	Построчное редактирование
3	Непосредственно прозрачный

`connptr` — указатель на обозначение присоединения к файлу `:CI.`. Присоединение должно быть назначено при помощи вызова `DQ$ATTACH`.

Выходной параметр

`exceptptr` — указатель на слово, в которое операционная система помещает код возврата.

Описание. Этот системный вызов изменяет режим, при помощи которого программа получает информацию с консольного входного устройства. При запуске операционной системы устанавливается режим построчного редактирования. Используя вызов `DQ$SPECIAL`, можно перейти от построчного редактирования к прозрачным режимам и назад к режиму построчного редактирования.

Режимы построчного редактирования следующие:

Прозрачный режим. Диалоговым программам часто требуется получить знаки с консоли в том виде, в котором они введены. В прозрачном режиме все знаки помещаются в буфер, заданный системным вызовом `DQ$READ`. (Исключениями являются знак `UC + C`, который прерывает программу, и знак `UC + D`, который не оказывает никакого воздействия на систему.) Операционная система возвращает управление вызывающей

программе при равенстве числа введенных знаков числу знаков, указанных в запросе чтения.

Режим построчного редактирования. Оператор за консолью имеет возможность исправлять ошибки ввода при помощи специальных клавиш, перед тем как прикладная программа получит введенные знаки.

Непосредственно прозрачный режим. Отличается от прозрачного режима тем, что операционная система возвращает управление программе непосредственно после вызова `DQ$READ` независимо от того, были или нет какие-либо знаки введены с момента последнего вызова `DQ$READ`. Если никаких знаков не было введено, то это будет указано в параметре `bytes$read` вызова `DQ$READ`. Знаки, введенные между двумя последовательными вызовами `DQ$READ`, хранятся в буфере предварительного ввода.

`DQ$SWITCH$BUFFER` подставляет новую командную строку вместо существующей:

```
char$offset=DQ$SWITCH$BUFFER (buff$ptr, except$ptr);
```

Входной параметр

`buff$ptr` — *указатель на строку, указывающую «новую» командную строку, аргументы которой должны быть возвращены при обращении к системному вызову `DQGETARGUMENT`.*

Выходные параметры:

`char$offset` — *слово, в которое УПИ помещает число, равное числу байтов с начала «старой» командной строки до последнего знака в последнем аргументе, обработанного вызовом `DQGETARGUMENT`. Иными словами, возвращаемое в параметре `char$offset` число указывает число старой командной строки, обработанной к моменту вызова;*

`except$ptr` — *указатель на слово, в которое операционная система помещает код возврата.*

О п и с а н и е. Когда программа вызывается с консоли, операционная система вызывает команду в буфер. Как правило, программа использует вызов `DQGETARGUMENT` для получения аргументов из команды. Если программа последовательно вызывает `DQ$READ` для получения дополнительных командных строк с консоли, то она может осуществить вызов `DQ$SWITCH$BUFFER` для указания буфера с новой командной строкой, из которой получают аргументы при вызове `DQGETARGUMENT`.

Можно использовать вызов `DQ$SWITCH$BUFFER` любое число раз для указания на различные строки в программе. Однако нельзя использовать вызов `DQ$SWITCH$BUFFER` для указания командной строки, которая вызвала программу, так как ее адрес известен операционной системе, а не программе пользователя. Поэтому необходимо использовать вызов `DQGETARGUMENT` для получения всех аргументов из строки вызова команды перед первым вызовом `DQ$SWITCH$BUFFER`.

Еще одна возможность использования вызова `DQ$SWITCH$BUFFER` — возврат места последнего аргумента в буфере. Следовательно, можно использовать данный вызов после одного системного вызова или более `DQGETARGUMENT` для определения в буфере начала следующего аргумента. Однако в этом случае происходит «сбор» буфера, т. е. очередной вызов `DQGETARGUMENT` вернет первый аргумент в буфе-

ре. Чтобы вернуться на нужную точку в буфере, из которой можно продолжать извлекать аргументы, повторно следует осуществить вызов `DQ$SWITCH$BUFFER`, но при этом использовать сумму стартового адреса буфера и значения, возвращенного в предыдущем вызове `DQ$SWITCH$BUFFER`.

DQ\$TRAP\$CC позволяет задать процедуру управления, если пользователь введет с консоли `УС + С`:

`CALL DQ$TRAP$CC (entry$ptr, except$ptr);`

Входной параметр

`entry$ptr` — *указатель* на точку входа процедуры обработки `УС + С`.

Выходной параметр

`except$ptr` — *указатель* на слово, в которое операционная система помещает код возврата.

Описание. Обычно, когда оператор вводит знак `УС + С` с консоли, система очищает буфер предварительного ввода и прерывает текущую исполняемую программу. Вызывая `DQ$TRAP$CC`, программа может назначить любую другую процедуру, которая автоматически получит управление при каждом вводе с консоли знака `УС + С`.

DQ\$TRAP\$EXCEPTION подставляет альтернативный драйвер обработки особых ситуаций вместо драйвера обработки особых ситуаций по умолчанию, предоставляемого операционной системой:

`CALL DQ$TRAP$EXCEPTION (address$ptr, except$ptr);`

Входной параметр

`address$ptr` — *указатель* на *указатель*, содержащий точку входа альтернативного драйвера обработки особых ситуаций.

Выходной параметр

`except$ptr` — *указатель* на слово, в которое операционная система помещает код возврата.

Описание. Как правило, драйвер обработки особых ситуаций прерывает программу, которая сделала вызов, приведший к особой ситуации, и выводит сообщение об этом на экран консоли. Вызов `DQ$TRAP$EXCEPTION` назначает альтернативный драйвер обработки особых ситуаций, которому будет передано управление при возникновении особой ситуации.

DQ\$TRUNCATE перемещает указатель конца файла на текущую позицию указателя присоединения именованного файла и, следовательно, освобождает часть файла, находящуюся за указателем:

`CALL DQ$TRUNCATE (connection, except$ptr);`

Входной параметр

`connection` — *обозначение* присоединения к именованному файлу данных, который должен быть «усечен». Усечение файла происходит с байта, определяемого указателем файла.

Выходной параметр

`except$ptr` — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Данный системный вызов приводит к усечению файла с текущей позиции и освобождает все пространство за указателем для его повторного занятия другими файлами. Если указатель расположен в конце или за концом файла, то усечение не выполняется; если указатель не находится в требуемом месте, то программа должна использовать системный вызов `DQ$SEEK` для позиционирования указателя перед выполнением системного вызова `DQ$TRUNCATE`.

Присоединение должно обладать правами записи, чтения или записи и чтения, установленными при открытии присоединения.

DQ\$WRITE копирует ряд байтов из буфера в файл:

`CALL DQ$WRITE (connection, buff$ptr, count, except$ptr)`

Входные параметры:

`connection` — *слово*, содержащее обозначение для присоединения к файлу, в который будет записана информация;

`buff$ptr` — *указатель* на буфер, содержащий последовательные байты, которые будут записаны в заданный файл;

`count` — *слово*, содержащее число байтов, которые будут записаны из буфера в файл.

Выходной параметр

`except$ptr` — *указатель на слово*, в которое операционная система помещает код возврата.

Описание. Системный вызов заставляет операционную систему поместить заданное число байтов из буфера в файл. Вызов `DQ$WRITE` начинает записывать с позиции, заданной указателем файла. После завершения операции записи указатель файла соответствует байту, следующему за последним записанным.

Если программа должна предварительно позиционировать указатель файла перед записью, то она может сделать это, используя системный вызов `DQ$SEEK`.

Заключение

Перспективы развития универсальных микроЭВМ



Развитие микроЭВМ СМ 1810 предполагается по трем основным направлениям:

- 1) разработка интеллектуальных модулей расширения;
- 2) построение сетей, в том числе локальных;
- 3) создание проблемно-ориентированного прикладного программного обеспечения.

Первое направление позволит увеличить производительность систем за счет децентрализации вычислительных ресурсов и повышения функциональных возможностей контроллеров. Предполагается создание интеллектуальных контроллеров и модулей на основе микропроцессоров и специализированных БИС, таких, как:

— коммуникационные контроллеры, обслуживающие 4—8 последовательных линий СТЫКА С2, посредством которых возможны как подсоединение терминалов, так и организация связей между ЭВМ. Создание подобных контроллеров позволит разгрузить основной процессор СМ 1810 (т. е. МЦП-16) от обслуживания последовательных интерфейсов в байтовом режиме, работа с линиями связи будет организовываться в блочном режиме с переносом таких функций, как предварительный ввод, первичное редактирование, формирование кадров, повторная передача данных и т. д., на контроллеры;

— модули УСО, позволяющие выполнять функции не только обычного дискретного и аналогового ввода — вывода, но и предварительной обработки (например, линеаризации), более сложных типов измерений (например, длительности дискретных импульсов), связи с новыми типами внешних устройств (например, выходящими на приборный интерфейс);

— видеоконтроллеры, выполняющие запросы высокого уровня, в том числе стандарты на машинную графику. Видеоконтроллеры получают широкое распространение в связи с расширением области применений в сторону обработки графической информации. Подобные видеоконтроллеры должны также повысить качество выводимой графической и псевдографической информации;

— узкоспециализированные процессоры, ориентированные на решение вычислительных задач, например быстрого преобразования Фурье.

Второе направление позволит реализовать сети, в первую очередь локальные. Предполагается наличие двух видов локальных сетей:

— для подключения контроллеров связи с объектом посредством интерфейса;

— для организации связи между микроЭВМ на основе протокола.

Локальные сети микроЭВМ типа СМ 1810 дают возможность организовать распределенное управление такими ресурсами, как внешняя память, высокоскоростные печатающие устройства. На базе локальных сетей возможно создание распределенных операционных систем, баз данных, реализация «электронной почты» и т. д. В ближайшее время значительная часть применений СМ 1810 в непромышленной сфере будет основываться на локальных сетях.

Третье направление позволит решить задачу интеграции пакетов прикладных программ с целью получения проблемно-ориентированных комплексов.

Массовый пользователь не должен быть специалистом по вычислительной технике, он должен на ее основе более эффективно решать свои задачи. А это возможно только за счет резкой переориентации прикладных программ на конечного получателя. Например, прикладное программное обеспечение для текстообработки должно быть доступно машинисткам, при этом оно должно оставлять возможности для внедрения приемов изготовления полиграфической продукции. В проблемно-ориентированных программных комплексах важное место займут средства для обучения, которые должны строиться по многоступенчатому принципу — от простого к сложному, давая возможность пользователю самому выбирать те предметные области, в которых ему необходимо обучение.

Наряду с эволюционным развитием и совершенствованием СМ 1810 предполагается создание новой модели

микроЭВМ данной линии. При ее разработке необходимо обеспечить:

- повышение пропускной способности систем за счет применения нового типа микропроцессора и использование более совершенного системного интерфейса;

- гибкую организацию параллельных вычислений в мультимикропроцессорных системах;

- повышение надежности системы путем дублирования важнейших узлов, применения встроенных средств диагностики, в том числе работающих при решении основных задач под управлением операционных систем;

- повышение степени унификации основных используемых программно-аппаратных решений для упрощения и повышения качества системного программного обеспечения;

- преемственность с микроЭВМ СМ 1810 для сохранения задела в области как технических, так и программных средств.

Рассмотрим некоторые основные принципы, заложенные в перспективную микроЭВМ.

Многомагистральная архитектура. Предполагается применение высокопроизводительного системного интерфейса, использующего несколько отдельных магистралей. При этом производительность повышается за счет параллельной работы магистралей, более эффективной реализации специализированных функций. Частичное дублирование функций (например, параллельных и последовательных магистралей) позволит строить отказоустойчивые системы. Арбитраж должен быть распределенным, что обязательно для построений многопроцессорных систем.

Пространство межсоединений. Данное пространство — особое адресное пространство на каждой плате микроЭВМ, которое содержит набор регистров, предназначенных для хранения идентификаторов плат, сведений о конфигурации, диагностической информации.

Пространство межсоединений упрощает задачу проектирования систем, их тестирования. Становятся возможными автоматическое определение текущей конфигурации системы и динамическая программно управляемая (от операционной системы) перенастройка модулей (плат), в том числе и на конкретное применение. Унифицируется и автоматизируется решение задачи тестирования. Например, модуль оперативной памяти может быть проверен в полном адресном пространстве без ручного пере-

ключения перемычек. Создается предпосылка для дистанционной диагностики, так как удаленным диагностическим программам доступна через регистры пространства межсоединений информация о состоянии и конфигурации компонентов конкретной системы.

Передача сообщений. В многопроцессорных системах существует межпроцессорный обмен информацией. В традиционной архитектуре обмен осуществляется путем использования разделяемой памяти (двухпортовой в процессорах и контроллерах либо общей интерфейсной). Недостаток этого принципа — резкое снижение общей производительности при увеличении числа запросов за счет издержек в программном обеспечении. Программы обслуживания запросов сложны для разработки и имеют низкую эффективность. Предполагается обработку сообщений производить в узле обслуживания автономно от процессора. Этот узел должен быть как в передающем, так и в принимающем модулях. Производительность возрастает по двум причинам:

1) вычислитель (т. е. процессор) освобождается от реализации протокола доступа к разделяемой памяти и может решать свои задачи;

2) обмен выполняется со скоростью магистрали, а не самого медленного процессора.

Рассмотрим магистрали, применяемые в новом системном интерфейсе перспективной модели микроЭВМ. Общая структура интерфейса в типичной конфигурации, приведенная на рис. 3к. 1, состоит из пяти магистралей:

— параллельная системная магистраль (высокопроизводительная универсальная 32-разрядная магистраль для обмена данными и межпроцессорных связей);

— локальная магистраль памяти (для обеспечения свободного от арбитража доступа к локальной памяти);

— последовательная системная магистраль (более дешевое средство для реализации тех же функций, которые выполняются параллельной системной магистралью);

— высокоскоростная магистраль ввода — вывода (для обмена данными между физически разнесенными периферийными устройствами и процессорами или контроллерами);

— локальная магистраль ввода — вывода (для подключения относительно простых устройств ввода — вывода).

Архитектура описываемой микроЭВМ требует наличия специального модуля для реализации общесистемных

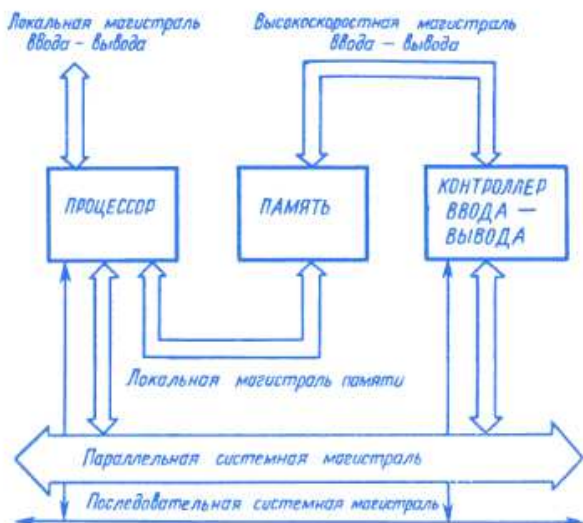


Рис. 3к.1. Общая структура интерфейса перспективной модели микроЭВМ

функций: генерации системных тактовых импульсов; информирования компонентов системы об аварии питания; начального сброса; назначения идентификаторов другим модулям; ведения астрономического времени суток.

Как видно, этот модуль является архитектурным развитием (в условиях нового интерфейса) модулей МСК и МСК-16 для СМ 1800 и СМ 1810 соответственно.

Важная особенность перспективной микроЭВМ — дальнейшее широкое внедрение стандартных (в первую очередь международных) интерфейсов для подключения внешних устройств.

Программное обеспечение описываемой микроЭВМ должно включать в себя следующие операционные системы:

- систему реального времени, совместимую с БОС 1810;
- систему общего назначения, совместимую с МДОС 1810;
- диалоговую единую мобильную систему (ДЕМОС).

В постоянной памяти центрального процессора должны размещаться начальные тесты, начальный загрузчик и отладочный системный монитор.

Операционная система реального времени должна обеспечивать следующие свойства: объектно-ориентированную архитектуру; многозадачность; обработку прерываний; диспетчерирование на основе приоритетов; многопрограммность и многопользовательский доступ; наличие нескольких систем ввода — вывода; иерархическое именование файлов во внешней памяти; независимость от устройств ввода — вывода; интерактивную конфигурируемость.

Эта система по отношению к БОС 1810 должна быть улучшена следующим образом:

- реализованы межпроцессорные взаимодействия для мультипроцессорных систем на основе прерываний, использующих передачу сообщений;

- обеспечена динамическая программно управляемая инициализация и идентификация всех модулей в системе;

- включены средства работы с общим (глобальным) счетчиком астрономического времени суток;

- введены средства для чтения и записи регистров межсоединений.

При этом будут реализованы дополнительные системные вызовы, применены новые объекты (например, порт для логического соединения процессоров).

Изменению подлежат следующие элементы операционной системы: ядро; базовая система ввода — вывода; драйверы устройств; интерфейс с оператором (только в части глобального времени суток); системный отладчик; начальный загрузчик; интерактивный конфигуратор.

Дальнейшее направление совершенствования операционной системы реального времени — динамическое распределение вычислительных процессов при работе системы.

Совместимость операционной системы общего назначения с МДОС 1810 позволит сохранить преемственность для большого числа пакетов прикладных программ, таких, как базы данных, текстовые процессоры, сетевые пакеты, средства машинной графики, интегрированные пакеты. Использование этой системы дает возможность обмениваться программами (и использовать их) с наиболее распространенными отечественными и зарубежными персональными ЭВМ.

Операционная система ДЕМОС реализуется на большинстве моделей СМ ЭВМ и на других типах ЭВМ. Основное назначение ее — инструментальное, с широким

использованием принципа мобильности. Таким образом достигается возможность обмена программ с различными ЭВМ, имеющими другую архитектуру и даже другую систему команд.

Система команд перспективной микроЭВМ будет расширением системы команд микропроцессора K1810BM86, используемого в СМ 1810, при этом станет возможной реализация простого переноса практически любых прикладных программ.

Уровни диагностики. Построение высоконадежных систем требует особого внимания к средствам диагностики. В общем случае используются три уровня диагностики:

- встроенная диагностика;
- тесты устройств и модулей (начальные и полные);
- системные диагностические тесты.

В перспективной микроЭВМ семейства СМ 1800 предполагается реализация всех трех уровней. Последние два уровня достаточно традиционны. Для реализации встроенных диагностических тестов на каждом модуле имеется однокристалльный микропроцессор, который диагностирует модуль при включении питания и по команде от диспетчера микроЭВМ.

Таким образом, основное применение перспективная универсальная микроЭВМ должна найти в качестве центральной станции локальных вычислительных сетей, где требуется наряду с высокими техническими характеристиками повышенная надежность.



В 1988 г. начат выпуск ряда новых модулей и комплексов микроЭВМ СМ 1810 с использованием советских малогабаритных жестких дисков типа «Винчестер» СМ 5514.

Основные параметры накопителя СМ 5514:

Диаметр диска — 130 мм

Неформатная емкость — до 26 М байт

Скорость передачи информации — 5 М бит/с

Число дорожек на поверхности диска — 320

Емкость сектора — от 128 до 1024 бит

Число головок — до 8

Состав новых модификаций комплексов приведен в табл. Зк.1.

Таблица 3к.1

Наименование составных частей комплекса	Комплексы СМ 1810				Основные характеристики
	12	22	32	42	
Модуль центрального процессора МЦП-16	1	1	1	1	Соответствует существующим МЦП-16 комплексов СМ 1810
Модуль центрального процессора МЦП-1	—	—	1	—	То же, для МЦП-1
Модуль системного контроля МСК-16	1	1	1	1	То же, для МСК-16
Контроллер НМД и НГМД СМ 1810.5126	1	1	1	1	Возможно подключение до двух жестких мини-дисков общим объемом памяти не менее 20 М байт и до четырех НГМД общим объемом памяти не менее 2 М байт
Накопитель на гибких магнитных дисках	1	1	—	1	Емкость не менее 0,5 М байт двухсторонний, 40 дорожек
Накопитель на жестком магнитном диске типа «Винчестер» СМ 5514	—	—	1	—	То же, 80 дорожек
Видеотерминал СМ 7209	1	1	1	1	Емкость не менее 10 М байт
Устройство печатающее алфавитно-цифровое СМ 6329	1	1	—	1	С выходом на интерфейс СТЫК С2
Модуль оперативный запоминающий СМ 1810.3515	—	—	1	—	ИРПС
Модуль оперативный запоминающий СМ 1810.3516.03	1	1	—	1	С выходом на интерфейс: ИРПР-М
Модуль связи с интерфейсами внешних устройств МСМ СМ 8501.02	—	—	1	—	ИРПС
Модуль сопряжения с ИРПС многоканальный СМ 1800.4106	2	2	1	—	Емкость 256 К байт
Операционная система ДОС 1810	—	—	—	1	Емкость 1 М байт с возможностью расширения до 4 М байт
Операционная система МИКРОС-86	—	1	—	1	4 канала связи с ИРПС или 4 канала связи с интерфейсом СТЫК С2
	—	—	1	—	4 канала связи с ИРПС
	1	—	—	—	—
	1	—	—	—	Аналог СР/М-86, занимает до 15 К байт, может работать с 1 М байт

Наименование составных частей комплекса	Комплексы СМ 1810				Основные характеристики
	12	22	32	42	
Операционная система БОС 1810	—	1	—	1	Аналог RMX-86
Тестовое обеспечение	1	1	1	1	—
Конструктивное исполнение					
— приборное	1	—	—	—	—
— тумба	—	1	—	—	С возможностью расширения одним блоком монтажным
— стойка	—	—	1	1	С возможностью расширения двумя блоками монтажными или одним блоком и кроссом
Наличие свободных мест для расширения:					
— с интерфейсом И41	2	1	6	4	—
— со специальным монтажом, предназначенным для установки плат ОЗУ емкостью 1 М байт	—	—	—	3	—

Краткие характеристики новых модулей семейства микроЭВМ СМ 1810 приводятся ниже.

Программатор СМ 1810.3708 предназначен для записи информации в микросхемы программируемых постоянных запоминающих устройств типа К573РФ2 и К573РФ4 с организацией 2 К бит \times 8 разрядов и К573РФ4, К573РФ6 с организацией 8 К бит \times 8 разрядов. В комплект принадлежностей программатора входит автономное устройство стирания информации типа УСИМ01.

Программатор может работать как с микроЭВМ СМ 1800, так и с микроЭВМ СМ 1810 под управлением одной из операционных систем ДОС 1810, БОС 1810 или МДОС 1810.

Энергонезависимый модуль оперативной памяти СМ 1810.3520 предназначен для хранения информации в управляющих комплексах семейства СМ 1800 при отключении напряжения сети. Для сохранения информации используется химический элемент А-316 «Прима».

Объем ОЗУ	— 16 К байт
Средний ток хранения	— 20 мкА
Рабочий ток потребления	— 2 А
Время выборки	— 500 нс

- Цикл записи слова и чтения — 550 мс
- Микросхема памяти — типа К537РУЗБ

Программируемый контроллер приборного интерфейса СМ 1810.9008 предназначен для автоматизации научных экспериментов и производственных процессов, имеющих выход на стандартный интерфейс IEEE448 (МЭК 625.1HP, GRIB, ПИ — приборный интерфейс, КОП — канал общего пользования).

- Система команд — в соответствии с архитектурой КР580ИК85
- Емкость памяти — 8 К байт ПЗУ и 56 К байт ОЗУ
- Минимальное время выполнения команд — 1,5 мкс
- Возможность адресации по КОП — до 961 приемника и до 961 источника информации

Программное обеспечение — резидентные программы, программы работы с приборами по алгоритму КОП

Устройство микропроцессорное дискретного ввода—вывода СМ 1810.9308 предназначено для ввода сигналов от дискретных датчиков в виде напряжения и «сухого контакта», вывода сигналов двухпозиционного управления, предварительной обработки информации и применения в составе управляющих вычислительных комплексов СМ 1810, СМ 1814. В системе команд устройства имеются 17 операций обработки однобитовых операндов. Это позволяет быстро манипулировать одноразрядовыми данными, непосредственно вычислять булевы функции, эффективно управлять вводом и выводом дискретной информации.

Устройство имеет шесть исполнений, отличающихся друг от друга количеством вводимых и выводимых сигналов. Во всех исполнениях имеются: блок элементов микропроцессорного контроллера, кабель, до шести кроссовых элементов (в зависимости от количества сигналов). Все каналы дискретного ввода могут быть инициативными, имеют гальваническую развязку на элементах, обеспечивающих допустимую амплитуду помехи общего вида до 500 В.

- Общее число программируемых каналов ввода — вывода — 53
- Диапазоны ввода сигналов напряжения — (6; 12; $24 \pm 20\%$) В
- Максимальное коммутируемое выходное напряжение — 28,8 В
- Максимальный коммутируемый выходной ток — 0,2 А
- Максимальная частота счета для число-импульсного ввода — 60,0 кГц
- Максимальная частота число-импульсного вывода — 20,0 кГц
- Длительность импульсов широтно-импульсного вывода — от 0,01 до 20 с
- Емкость оперативной памяти — 2 К байт
- Емкость полупостоянной памяти — 16 К байт
- Габаритные размеры блока микропроцессорного контроллера — 246×237,5×16 мм
- Габаритные размеры кроссовых блоков элементов — 118×220×37 мм
- Масса устройства — не более 4 кг
- Число каналов число-импульсного ввода — 2

Число каналов число-импульсного вывода (широотно-импульсного)	— 1
Разрядность число-импульсного ввода — вывода	— 16

Кроссовые блоки элементов позволяют подключать одновременно до 48 программируемых каналов ввода — вывода.

Контроллер видеографический (ВГК) СМ 1810.7005 преобразует закодированную символьную и графическую информацию в видеосигнал для отображения на экранах телевизионных индикаторов; формирует полный телевизионный сигнал для черно-белого индикатора, сигналы признаков цветности R, G, B и сигнал синхронизации для цветных индикаторов; обеспечивает следующие форматы изображения:

СМ 1810.7005.01 — 750×512 точек при частоте строчной развертки 15,65 кГц;

СМ 1810.7005.02 — 640×512 точек при частоте строчной развертки 27,0 кГц;

СМ 1810.7005.03 — 1024×768 точек при частоте строчной развертки 20,4 кГц.

Система команд ВГК позволяет:

— вычерчивать на экране точки, линии, дуги, прямоугольники, заполненные зоны, символы, наклонные символы; максимальный размер знакоместа символа 8×8 точек;

— задавать тип линий (сплошная, штриховая, штрихпунктирная и т. д.) при построении линии, дуги, прямоугольника;

— выводить на экране массив байтов;

— воспроизводить монохромные, полутонные и цветные изображения;

— увеличивать, панорамировать и создавать полиэкранные изображения;

— программировать формат изображения;

— выводить изображение 16 различными цветами (уровнями яркости) из палитры 4096 цветов.

Контроллер видеографический обеспечивает сопряжение с черно-белыми и цветными телевизионными индикаторами в соответствии с табл. Эк.2, а также с серийно выпускаемыми прикладными телевизионными установками и видеоконтрольными устройствами, совместимыми по входным сигналам.

Видеоконтроллер цветной (ВКЦ) СМ 1810.7006 преобразует символьную и графическую информацию в видеосигнал для отображения на модуле индикации М32Ц1/М32Ц2 и других видеоконтрольных устройствах, совместимых с ним.

ВКЦ обеспечивает следующие параметры:

Максимальное число символов в строке — 80

Максимальное число строк — 25

Набор символов — 256

Число цветов символов — 16

Число цветов фона — 8

Управление — яркостью, мерцанием, инверсией

Разрешающая способность точек \times цвет в графическом режиме —
 $(640 \times 200) \times 1$
 $(320 \times 200) \times 4$
 $(160 \times 100) \times 16$

Модуль системного контроля МСК52 СМ 1810.2010 помимо функций, выполняемых модулем МСК16, обеспечивает

Таблица 3к.2

Наименование и условное обозначение телевизионного индикатора	Исполнение ВГК		
	СМ 1810 7005 01	СМ 1810 7005 02	СМ 1810 7005 03
Модуль индикации черно-белый А543-13	+	—	—
Модуль индикации цветной А543-14	+	—	—
Модуль индикации цветной М32Ц11	+	—	—
Модуль индикации цветной М32Ц31	—	+	—
Модуль индикации цветной М51Ц21	—	—	+
Видеоконтрольное устройство ВК42Ц6 и другие устройства, совместимые по входному сигналу	+	—	—

Примечание. Знак «+» означает, что сопряжение обеспечивается, знак «—» — что сопряжение не обеспечивается.

связь с клавиатурой и совместно с модулем ВКЦ позволяет создавать на базе семейства СМ 1810 комплексы, совместимые по программному обеспечению с персональными ЭВМ типа ЕС 1841.

В составе семейства микроЭВМ СМ 1800 предполагается использовать модули индикации цветные повышенной разрешающей способности, основные технические характеристики которых приведены в табл. 3к.3.

Таблица 3ж.3

Наименование параметра	Норма параметра для модулей						
	M32Ц1	M32Ц2	M32Ц3	M42Ц1	M51Ц1	M52Ц2	M51Ц3
Номинальные размеры изображения							
— по горизонтали В, мм	230	230	230	300	340	340	340
— по вертикали Н, мм	170	170	170	230	250	250	250
Разрешающая способность по горизонтали (линий)							
— в пределах круга диаметром 0,8Н (зона 1)	640	640	640	550	1024	1024	1024
— в остальной части изображения (зона 2)	600	600	600	500	850	850	850
Разрешающая способность по вертикали (линий)							
— в зоне 1	576	600	600	500	576	768	800
— в зоне 2	500	500	500	450	576	700	700
Яркость изображения, кд/м ²	50	50	50	80	50	50	50
Контрастность изображения	1:50	1:50	1:50	1:50	1:50	1:50	1:50
Наличие антибликового покрытия экрана	Есть	Есть	Есть	Есть	Есть	Есть	Есть
Мощность, потребляемая модулем, Вт	80	85	100	80	85	90	110
Масса модуля, кг	16	16	16	22	28	28	28
Габаритные размеры модуля, мм	320 × 300 × 400	320 × 300 × 400	320 × 300 × 400	430 × 370 × 440	620 × 550 × 550	620 × 550 × 550	620 × 550 × 550

Список литературы

1. OEM SYSTEM HANDBOOK INTEL, 1985.
 2. МикроЭВМ. Архитектура, программирование, применение/ А. В. Гиглавый, Н. Д. Кабанов, Н. Л. Прохоров, А. Н. Шкамарда. — М.: Финансы и статистика, 1984.
 3. P. Norton, Inside the IBM PC. A Prentice — Hall Publishing and Communications Company, 1983.
 4. Крамфус И. Р., Новик А. Г., Перцов Е. Е. Обзор операционной системы реального времени RMX/80. — В сб.: Операционные системы для микроЭВМ. — М.: МЦНТИ, 1984, с. 12—85.
 5. СМ ЭВМ: Комплексование и применение/ Под ред. Н. Л. Прохорова. — М.: Финансы и статистика, 1986.
 6. Иткин Л. К. Программное обеспечение СМ 1800 линии ОС 1800. — В сб.: Микро-ЭВМ СМ 1800. Вопросы применения, ТС-12. — М.: ЦНИИТЭИприборостроения, 1983, с. 7—8.
 7. МикроДОС. Мобильная операционная система для микроЭВМ. Ч. I, II и III./ Под ред. Е. Е. Дудникова. — М.: МЦНТИ, 1985.
 8. Система обработки экономической информации на малых ЭВМ/ Н. Д. Кабанов, М. Р. Ковтун, Б. В. Лукьянов и др. — М.: Книга, 1981.
 9. Бабанов И. И., Хахалина Л. В. Библиотека программ общего назначения для СМ 1800. — В сб.: Технические и программные средства СМ 1800. Средства межмашинного обмена и системное применение. — Труды ИНЭУМа, вып. 105, 1984, с. 31—34.
 10. Абрамович С. Н., Бойко В. В., Бутрин Б. П. и др. Профессиональные персональные ЭВМ «Искра-226». Микропроцессорные средства и системы, 1985, № 2.
 11. Южаков А. Математическое обеспечение ЭВМ «Искра-226». Пакеты прикладных программ для микро-ЭВМ «Искра-226». — М.: Международный центр научной и технической информации, 1985, вып. 28.
 12. Жучков А. А. Система подготовки и отладки программного обеспечения для телекоммуникационного интерфейса ЭВМ «Искра-226». — М.: Международный центр научной и технической информации, 1985, вып. 28.
 13. ИПС ПРИМА/Т. Бабаян, Ю. Горностаев, С. Зиновьев, В. Илларионов. — М.: Международный центр научной и технической информации, 1985, вып. 28.
 14. Горностаев Ю., Пасько А., Пилюгин В., Сурина О. Система машинной графики САГРАФ. — М.: Международный центр научной и технической информации, 1985, вып. 28.
 15. Горностаев Ю., Зиновьев С., Южаков А. Интеллектуальный терминал ИТ «Искра-226» для работы с базами данных научно-технической информации, 1985, вып. 28.
-

Оглавление

Введение	5	
Глава 1		
Принципы построения семейства микроЭВМ СМ 1800		
1.1. Элементная база	7	
1.2. Архитектура микроЭВМ	14	
1.3. Базовые модули СМ 1810	26	
1.4. Конструкция и варианты конструктивного исполнения микроЭВМ	39	
Глава 2		
Программное обеспечение семейства микроЭВМ СМ 1800		
2.1. Операционные системы общего назначения	44	
2.2. Инструментальные системы	53	
2.3. Операционные системы реального времени	64	
2.4. Базовое прикладное программное обеспечение	82	
Глава 3		
Построение комплексов моделей семейства СМ 1800 и их применение		
3.1. Формирование комплексов	86	
3.2. Примеры построения комплексов	91	
3.3. Области применения	94	
Глава 4		
МикроЭВМ «Искра-226»		
4.1. Архитектура микроЭВМ	102	
4.2. Основные блоки микроЭВМ	106	
4.3. Интерфейс ввода—вывода информации	109	
4.4. Системное программное обеспечение	117	
4.5. Входной язык	118	
4.6. Пакеты прикладных программ	119	
Приложение		
Универсальный программный интерфейс		
П.1. Системные вызовы управления памятью	123	
П.2. Системные вызовы управления файлами	123	
П.3. Коды возврата и системные вызовы обработки особых ситуаций	124	
П.4. Вызовы УПИ из программ	124	
П.5. Описание системных вызовов	125	
Заключение. Перспективы развития универсальных микроЭВМ		144
Список литературы	157	

МикроЭВМ

В 8-ми книгах

НИКОЛАЙ ДМИТРИЕВИЧ КАБАНОВ
ВЛАДИМИР САМОЙЛОВИЧ КРАВЧЕНКО
АЛЕКСАНДР НИКОЛАЕВИЧ ШКАМАРДА
ДМИТРИЙ ИВАНОВИЧ ПАНФИЛОВ

Кн. 6

Универсальные машины
семейства СМ 1800

Заведующая редакцией Н. И. Хрусталева
Редактор С. М. Оводова
Мл. редакторы Г. Г. Бучина, Е. В. Судьенкова
Художник В. М. Боровков
Художественный редактор В. И. Мешалкин
Технический редактор Г. А. Фетисова
Корректор Г. И. Кострикова

ИБ 7153

Изд. № СТД—585. Сдано в набор 20.10.87. Подп. в печать 30.06.88.
Т-15510. Формат 84×108¹/₃₂. Бум. офс. № 1. Гарнитура Литературная.
Печать офсетная. Объем 8,40 усл. печ. л. 17,22 усл. кр.-отт. 8,81 уч.-изд. л.
Тираж 150 000 экз. Зак. № 1001. Цена 55 коп.

Издательство «Высшая школа», 101430, Москва, ГСП-4, Неглинная ул., д. 29/14.

Ярославский полиграфкомбинат Союзполиграфпрома при Государственном комитете СССР по делам издательства, полиграфии и книжной торговли. 150014, Ярославль, ул. Свободы, 97.

**В 1989 г. издательством
будет выпущено учебное пособие**

К. А. Иыуду «Надежность, контроль и диагностика вычислительных машин и систем», 15 л.

В пособии изложены вопросы оценки и расчета надежности вычислительных машин и систем на основе статистических, структурных и эксплуатационных моделей; вопросы надежности программного обеспечения; методы контроля и диагностирования вычислительных машин и систем; методы построения отказоустойчивых вычислительных систем.

Предназначается для студентов вузов, обучающихся по специальности «Вычислительные машины, комплексы, системы и сети». Могут пользоваться студенты технических вузов, а также инженерно-технические работники.

Уважаемые читатели!

По вопросам приобретения учебного пособия просим обращаться в местные отделения Книготорга или книжные магазины по месту жительства. (См. план выпуска литературы для вузов и техникумов, 1989 г., поз. 191).

ИЗДАТЕЛЬСТВО «ВЫСШАЯ ШКОЛА»



СЕРИЯ КНИГ

ИЗДАТЕЛЬСТВО
"ВЫСШАЯ ШКОЛА"

- 1** Семейство ЭВМ
«Электроника 60»
- 2** Персональные
ЭВМ
- 3** Семейство ЭВМ
«Электроника К1»
- 4** Управляющие
системы
«Электроника НЦ»
- 5** Персонально-
профессиональные
ЭВМ
- 6** Универсальные
машины
семейства СМ1800
- 7** Учебные стенды
- 8** МикроЭВМ
в учебных
заведениях

