

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального образования
“ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

С. Н. Ливенцов, А. Д. Вильнин, А. Г. Горюнов

**ОСНОВЫ
МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ**

Учебное пособие

Издательство Томского политехнического университета
Томск 2007

УДК 681.322

Л55

Ливенцов С. Н.

Л55

Основы микропроцессорной техники: учебное пособие / С. Н. Ливенцов, А. Д. Вильнин, А. Г. Горюнов. – Томск: Изд-во Томского политехнического университета, 2007. – 118 с.

Учебное пособие посвящено вопросам микропроцессорной техники, содержит основные понятия и классификацию микропроцессоров и состоит из трех глав. В первой главе рассмотрена архитектура типового микропроцессора, во второй – память микропроцессорной системы, а в третьей – организация ввода/вывода в микропроцессорной системе.

Пособие подготовлено на кафедре «Электроника и автоматика физических установок» ТПУ и предназначена для студентов очного обучения специальности 140306.

УДК 681.322

Рекомендовано к печати Редакционно-издательским советом
Томского политехнического университета

Рецензенты

Директор учебно-научного инновационного центра
радиотехнического факультета Томского государственного
университета, доктор физико-математических наук, профессор
О. П. Толбанов

Доктор технических наук, первый проректор
Северской государственной технологической академии,
профессор
Б. М. Кербель

© Томский политехнический университет, 2007

© Оформление. Издательство ТПУ, 2007

Оглавление

ВВЕДЕНИЕ	5
1. Основные понятия	8
2. Классификация микропроцессоров.....	12
3. Архитектура микропроцессора	18
3.1. Понятие архитектуры микропроцессора	18
3.2. Основные характеристики микропроцессора	19
3.3. Типы архитектур микропроцессоров	20
3.4. Архитектурно-функциональные принципы построения ЭВМ	25
3.5. Структура типовой ЭВМ (персонального компьютера).....	26
3.6. Система команд микропроцессора.....	42
3.7. Структура микропроцессора.....	48
4. Память микропроцессорной системы.....	64
4.1. Основные характеристики полупроводниковой памяти.....	67
4.2. Постоянные запоминающие устройства.....	68
4.3. Оперативные запоминающие устройства.....	71
4.4. Запоминающие устройства с произвольной выборкой.....	75
4.5. Микросхемы памяти в составе микропроцессорной системы	76
4.6. Буферная память	77
4.7. Стековая память	80
5. Организация ввода/вывода в микропроцессорной системе	82
5.1. Программная модель внешнего устройства	82
5.2. Форматы передачи данных	85
5.3. Параллельная передача данных.....	89
5.4. Последовательная передача данных	94
5.5. Способы обмена информацией в микропроцессорной системе	100
Библиографический список	117

СПИСОК СОКРАЩЕНИЙ

- АЛУ – арифметико-логическое устройство;
БИС – большая интегральная схема;
ВАХ – вольт-амперная характеристика;
ВВ – ввод/вывод;
ВЗУ – внешнее запоминающее устройство;
ВУ – внешнее устройство;
ДОЗУ – динамические оперативные запоминающие устройства;
ЗУ – запоминающее устройство;
ЗУПВ – запоминающее устройство с произвольной выборкой;
ИМС – интегральная микросхема;
КМОП – комплиментарный металл – оксид – полупроводник;
КОП – код операции;
МДОП – металл–диэлектрик–оксид–полупроводник;
МДП – металл–диэлектрик–полупроводник;
МНОП – металл – нитрид кремния – оксид кремния – полупроводник;
МОП – металл–оксид–полупроводник;
МП – микропроцессор;
МПС – микропроцессорная система;
МТ – микропроцессорная техника;
ОЗУ – оперативное запоминающее устройство;
ПВВ – порт ввода/вывода;
ПДП – прямой доступ к памяти;
ПЗУ – постоянное запоминающее устройство;
ПК – персональный компьютер;
ПКП – программируемый контроллер прерываний;
ППЗУ – программируемые постоянные запоминающие устройства;
РОН – регистр общего назначения;
СОЗУ – статические оперативные запоминающие устройства;
ТТЛ – транзисторно-транзисторная логика;
УВВ – устройство ввода/вывода;
УУ – устройство управления;
ЦП – центральный процессор;
ША – шина адреса;
ШД – шина данных;
ШУ – шина управления;
ЭВМ – электронно-вычислительная машина;
CISC – Complex Instruction Set Computing (микропроцессор с полным набором команд);
RISC – Reduced Instruction Set Computing (микропроцессор с сокращенным набором команд).

ВВЕДЕНИЕ

ЭВМ получили широкое распространение, начиная с 50-х годов. Прежде это были очень большие и дорогие устройства, используемые лишь в государственных учреждениях и крупных фирмах. Размеры и форма цифровых ЭВМ неузнаваемо изменились в результате разработки новых устройств, называемых микропроцессорами.

Микропроцессор (МП) – это программно-управляемое электронное цифровое устройство, предназначенное для обработки цифровой информации и управления процессом этой обработки, выполненное на одной или нескольких интегральных схемах с высокой степенью интеграции электронных элементов.

В 1970 году Маршиан Эдвард Хофф из фирмы Intel сконструировал интегральную схему, аналогичную по своим функциям центральному процессору большой ЭВМ, – первый микропроцессор Intel-4004, который уже в 1971 году был выпущен в продажу.

15 ноября 1971 г. можно считать началом новой эры в электронике. В этот день компания приступила к поставкам первого в мире микропроцессора Intel 4004.

Это был настоящий прорыв, ибо МП Intel-4004 размером менее 3 см был производительнее гигантской машины ENIAC. Правда работал он гораздо медленнее и мог обрабатывать одновременно только 4 бита информации (процессоры больших ЭВМ обрабатывали 16 или 32 бита одновременно), но и стоил первый МП в десятки тысяч раз дешевле. Кристалл представлял собой 4-разрядный процессор с классической архитектурой ЭВМ гарвардского типа и изготавливался по передовой р-канальной МОП-технологии с проектными нормами 10 мкм. Электрическая схема прибора насчитывала 2300 транзисторов. МП работал на тактовой частоте 750 кГц при длительности цикла команд 10,8 мкс. Чип i4004 имел адресный стек (счетчик команд и три регистра стека типа LIFO), блок РОНов (регистры сверхоперативной памяти или регистровый файл – РФ), 4-разрядное параллельное АЛУ, аккумулятор, регистр команд с дешифратором команд и схемой управления, а также схему связи с внешними устройствами. Все эти функциональные узлы объединялись между собой 4-разрядной ШД. Память команд достигала 4 кб (для сравнения: объем ЗУ миниЭВМ в начале 70-х годов редко превышал 16 кб), а РФ ЦП насчитывал шестнадцать 4-разрядных регистров, которые можно было использовать и как восемь 8-разрядных. Такая организация РОНов сохранена и в последующих МП фирмы Intel. Три регистра стека обеспечивали три уровня вложения подпрограмм. МП i4004 монтировался в пластмассовый или ме-

таллокерамический корпус типа DIP (Dual In-line Package) всего с 16 выводами.

В систему его команд входило всего 46 инструкций. Вместе с тем кристалл располагал весьма ограниченными средствами ввода/вывода, а в системе команд отсутствовали операции логической обработки данных (И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ), в связи с чем их приходилось реализовывать с помощью специальных подпрограмм. Модуль i4004 не имел возможности останова (команды HALT) и обработки прерываний.

Цикл команды процессора состоял из 8 тактов задающего генератора. Была мультиплексированная ША (шина адреса)/ШД (шина данных), адрес 12-разрядный передавался по 4 разряда.

1 апреля 1972 г. фирма Intel начала поставки первого в отрасли 8-разрядного прибора i8008. Кристалл изготавливался по р-канальной МОП-технологии с проектными нормами 10 мкм и содержал 3500 транзисторов. Процессор работал на частоте 500 кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора).

В отличие от своих предшественников МП имел архитектуру ЭВМ принстонского типа, а в качестве памяти допускал применение комбинации ПЗУ и ОЗУ.

По сравнению с i4004 число РОН уменьшилось с 16 до 8, причем два регистра использовались для хранения адреса при косвенной адресации памяти (ограничение технологии – блок РОН аналогично кристаллам 4004 и 4040 в МП 8008 был реализован в виде динамической памяти). Почти вдвое сократилась длительность машинного цикла (с 8 до 5 состояний). Для синхронизации работы с медленными устройствами был введен сигнал готовности READY.

Система команд насчитывала 65 инструкций. МП мог адресовать память объемом 16 кб. Его производительность, по сравнению с четырехразрядными МП, возросла в 2,3 раза. В среднем для сопряжения процессора с памятью и устройствами ввода/вывода требовалось около 20 схем средней степени интеграции.

Возможности р-канальной технологии для создания сложных высокопроизводительных МП были почти исчерпаны, поэтому "направление главного удара" перенесли на n-канальную МОП-технологию.

1 апреля 1974 МП Intel 8080 был представлен вниманию всех заинтересованных лиц. Благодаря использованию технологии п-МОП с проектными нормами 6 мкм, на кристалле удалось разместить 6 тыс. транзисторов. Тактовая частота процессора была доведена до 2 МГц, а длительность цикла команд составила уже 2 мкс. Объем памяти, адресуемой процессором, был увеличен до 64 кб. За счет использования

40-выводного корпуса удалось разделить ША и ШД, общее число микросхем, требовавшихся для построения системы в минимальной конфигурации сократилось до 6.

В РФ были введены указатель стека, активно используемый при обработке прерываний, а также два программнонедоступных регистра для внутренних пересылок. Блок РОНов был реализован на микросхемах статической памяти. Исключение аккумулятора из РФ и введение его в состав АЛУ упростило схему управления внутренней шиной.

Новое в архитектуре МП – использование многоуровневой системы прерываний по вектору. Такое техническое решение позволило довести общее число источников прерываний до 256 (до появления БИС контроллеров прерываний схема формирования векторов прерываний требовала применения до 10 дополнительных чипов средней интеграции). В i8080 появился механизм прямого доступа в память (ПДП) (как ранее в универсальных ЭВМ IBM System 360 и др.).

ПДП открыл зеленую улицу для применения в микроЭВМ таких сложных устройств, как накопители на магнитных дисках и лентах дисплеи на ЭЛТ, которые и превратили микроЭВМ в полноценную вычислительную систему. Традицией компании, начиная с первого кристалла, стал выпуск не отдельного чипа ЦП, а семейства БИС, рассчитанных на совместное использование.

1. Основные понятия

Микропроцессор (МП) – это микросхема или совокупность небольшого числа микросхем (соответственно один или несколько кристаллов БИС), выполняющая над данными арифметические и логические операции и осуществляющая программное управление вычислительным процессом [1, 2, 3, 4].

Микропроцессорные средства выпускаются промышленностью в виде наборов микросхем (chip-set) совместимых по уровням напряжения питания, сигналам и представлению информации, включающих МП, микросхемы оперативной и постоянной памяти, управления вводом/выводом, генератора тактовых сигналов и др.

Микропроцессоры (микропроцессорные средства) служат основой для создания различных универсальных и специализированных микро-ЭВМ, микропроцессорных информационно-управляющих систем, программируемых микроконтроллеров, разнообразных микропроцессорных приборов и устройств контроля, управления и обработки данных.

Микро-ЭВМ или микрокомпьютером называют устройство обработки данных, содержащее один или несколько микропроцессоров, БИС постоянной и оперативной памяти, БИС управления вводом и выводом информации и некоторые другие схемы [1, 3, 4]. Микрокомпьютер такого состава иногда называют «голым» из-за отсутствия в нем периферийных устройств (внешних ЗУ и устройств ввода и вывода информации). Микрокомпьютеры в такой конфигурации часто применяют в качестве встраиваемых в различные станки, машины, технологические процессы управляющих устройств (контроллеров).

Микрокомпьютеры широкого назначения, используемые для выполнения вычислительных работ, управления сложными технологическими процессами, оснащаются необходимыми периферийными устройствами (дисплеями, печатающими устройствами, ЗУ на гибких дисках, аналого-цифровыми и цифроаналоговыми преобразователями и др.).

Микропроцессорной системой (МП-системой) обычно называют специализированную информационную или управляющую систему, построенную на основе микропроцессорных средств [1, 3, 4].

Микрокомпьютер с небольшими вычислительными ресурсами и упрощенной системой команд, ориентированный не на производство вычислений, а на выполнение процедур логического управления различным оборудованием, называют программируемым микроконтроллером или просто **микроконтроллером** [1,4].

Логическая организация (архитектура) микропроцессоров (микропроцессорных средств) ориентирована на достижение универсальности применения, высокой производительности и технологичности.

Универсальность МП (микропроцессорных средств) определяется возможностью их разнообразного применения и обеспечивается программным управлением МП, позволяющим производить программную настройку МП на выполнение определенных функций, магистрально-модульным принципом построения, а также специальными аппаратурно-логическими средствами: сверхоперативной регистровой памятью, многоуровневой системой прерывания, прямым доступом к памяти, программно-настраиваемыми схемами управления вводом/выводом и т. п.

Относительно высокая производительность МП достигается использованием для их построения быстродействующих больших и сверхбольших интегральных электронных схем и специальных архитектурных решений, таких как стековая память, разнообразные способы адресации, гибкая система команд (или микрокоманд) и др.

Технологичность микропроцессорных средств обеспечивается модульным принципом конструирования, который предполагает реализацию этих средств в виде набора функционально законченных БИС, просто объединяемых в соответствующие вычислительные устройства, машины и комплексы.

Высокая универсальность и гибкость МП, достигаемая благодаря программному управлению, низкая стоимость, небольшие размеры, повышенная надежность, возможность встраивания микропроцессорных средств в приборы, машины и технологические процессы, обеспечивают микропроцессорам исключительно широкое применение в различных управляющих и обрабатывающих данные цифровых устройствах и системах.

Использование микропроцессоров приводит к изменению характера проектной работы разработчика устройств и систем автоматики: во многих случаях проектирование схем заменяется разработкой программ настройки микропроцессорной аппаратуры на выполнение определенных функций.

При разработке средств микропроцессорной техники нашли дальнейшее, более глубокое развитие следующие принципы: модульность; магистральность; микропрограммируемость; регулярность структуры.

Модульная организация предполагает построение систем на основе набора модулей конструктивно, функционально и электрически законченных устройств, позволяющих самостоятельно или в совокуп-

ности с другими модулями решать вычислительные или управленческие задачи определенного класса [5, 6].

Модульный подход способствует стандартизации элементов все более высоких уровней и сокращению затрат на проектирование систем, а также упрощает наращивание мощности и реконфигурацию систем, отодвигает время морального старения технических средств.

Многофункциональность (универсальность) и специализация модулей – эти два противоположных качества придают модулям в процессе компромиссных решений для различных классов систем, исходя из обеспечения соответствия структуры системы характеру выполняемых задач.

Целесообразно создавать системы в виде совокупности многофункциональных и специализированных модулей, проблемно и функционально ориентированных в рамках определенных классов задач, алгоритмов, функций.

Магистральность – способ обмена информацией внутри модулей и между модулями с помощью упорядоченных связей (в отличие от произвольных связей, реализующих принцип «каждый с каждым»), минимизирующий число связей. Обмен осуществляется с помощью общих магистралей (шин), объединяющих входные и выходные линии отдельных элементов и модулей. Магистральность – один из способов обеспечения регулярности структуры системы и стандартизации интерфейсов. С технической точки зрения – это способ обмена в виде создания специальных двунаправленных буферных каскадов с тремя устойчивыми состояниями и использования временного мультиплексирования каналов обмена [5, 6].

Микропрограммируемость – способ организации управления, позволяющий осуществить проблемную ориентацию системы. Микропрограммируемость повышает гибкость устройств (за счет возможности смены микропрограмм), увеличивает регулярность их структуры (за счет широкого использования матричных структур типа памяти), повышает надежность устройств (за счет применения серийно освоенных БИС памяти), упрощает контроль функционирования устройства (за счет того, что контроль блока микропрограммного управления сводится, по существу, к контролю содержимого ЗУ).

Регулярность структуры – предполагает закономерную повторяемость элементов структуры и связей между ними; регулярность структуры системы следует рассматривать на различных уровнях её организации.

В целом можно отметить, что МП как программируемое цифровое устройство обработки информации характеризуется следующими по-

казателями и связями их с внутренней и внешней структурой МП: разрядность; емкость адресуемой памяти; универсальность (специализация); число внутренних регистров; магистральность; микропрограммное управление; возможность и количество уровней прерывания; наличие стековой организации памяти и количество стековых регистров; наличие и состав резидентного и кросс-программного обеспечения.

Микропроцессорная автоматическая система (МПАС) – это автоматическая система со встроенными в нее средствами микропроцессорной техники (МП) [5, 6].

2. Классификация микропроцессоров

1. По числу больших интегральных схем (БИС) в микропроцессорном комплекте различают *однокристалльные, многокристалльные и многокристалльные секционные* [1, 2, 3] микропроцессоры.

Процессоры даже самых простых ЭВМ имеют сложную функциональную структуру, содержат большое количество электронных элементов и множество разветвленных связей. Изменять структуру процессора необходимо так, чтобы полная принципиальная схема или ее части имели количество элементов и связей, совместимое с возможностями БИС. При этом микропроцессоры приобретают внутреннюю магистральную архитектуру, т. е. в них к единой внутренней информационной магистрали подключаются все основные функциональные блоки (арифметико-логический, рабочих регистров, стека, прерываний, интерфейса, управления и синхронизации и др.).

Для обоснования классификации микропроцессоров по числу БИС надо распределить все аппаратные блоки процессора между основными тремя функциональными частями: операционной, управляющей и интерфейсной. Сложность операционной и управляющей частей процессора определяется их разрядностью, системой команд и требованиями к системе прерываний; сложность интерфейсной части разрядностью и возможностями подключения других устройств ЭВМ (памяти, внешних устройств, датчиков и исполнительных механизмов и др.). Интерфейс процессора содержит несколько десятков информационных шин данных (ШД), адресов (ША) и управления (ШУ).

Однокристалльные микропроцессоры получаются при реализации всех аппаратных средств процессора в виде одной БИС или СБИС (сверхбольшой интегральной схемы). По мере увеличения степени интеграции элементов в кристалле и числа выводов корпуса параметры однокристалльных микропроцессоров улучшаются. Однако возможности однокристалльных микропроцессоров ограничены аппаратными ресурсами кристалла и корпуса. Для получения многокристалльного микропроцессора необходимо провести разбиение его логической структуры на функционально законченные части и реализовать их в виде БИС (СБИС). Функциональная законченность БИС многокристалльного микропроцессора означает, что его части выполняют заранее определенные функции и могут работать автономно.

На рис. 1,а показано функциональное разбиение структуры процессора при создании трехкристалльного микропроцессора (пунктирные линии), содержащего БИС операционного (ОП), БИС управляющего (УП) и БИС интерфейсного (ИП) процессоров.

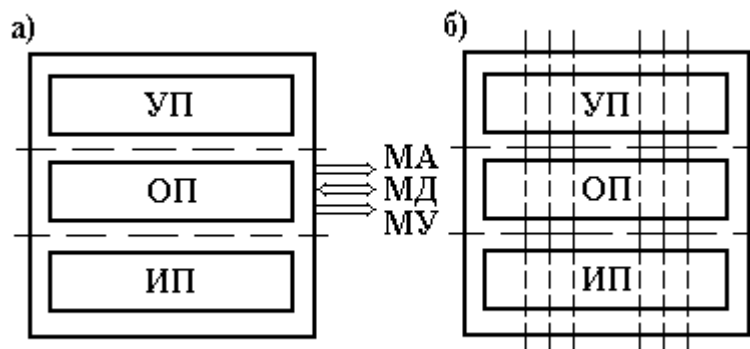


Рис. 1. Функциональная структура процессора (а) и ее разбиение для реализации процессора в виде комплекта секционных БИС.

Операционный процессор служит для обработки данных, управляющий процессор выполняет функции выборки, декодирования и вычисления адресов операндов и также генерирует последовательности микрокоманд. Автономность работы и большое быстродействие УП позволяет выбирать команды из памяти с большей скоростью, чем скорость их исполнения ОП. При этом в УП образуется очередь еще не исполненных команд, а также заранее подготавливаются те данные, которые потребуются ОП в следующих циклах работы. Такая опережающая выборка команд экономит время ОП на ожидание операндов, необходимых для выполнения команд программ. Интерфейсный процессор позволяет подключить память и периферийные средства к микропроцессору; он, по существу, является сложным контроллером для устройств ввода/вывода информации. Интерфейсный процессор выполняет также функции канала прямого доступа к памяти.

Выбираемые из памяти команды распознаются и выполняются каждой частью микропроцессора автономно и поэтому может быть обеспечен режим одновременной работы всех интегральных схем МП, т. е. конвейерный поточный режим исполнения последовательности команд программы (выполнение последовательности с небольшим временным сдвигом). Такой режим работы значительно повышает производительность микропроцессора.

Многокристалльные секционные микропроцессоры получают в том случае, когда в виде БИС реализуются части (секции) логической структуры процессора при функциональном разбиении ее вертикальными плоскостями (рис. 1,б). Для построения многоразрядных микропроцессоров при параллельном включении секций БИС в них добавляются средства "стыковки".

Для создания высокопроизводительных многоразрядных микропроцессоров требуется столь много аппаратных средств, не реализуемых в доступных БИС, что может возникнуть необходимость еще и в функциональном разбиении структуры микропроцессора горизонтальными плоскостями. В результате рассмотренного функционального разделения структуры микропроцессора на функционально и конструктивно законченные части создаются условия реализации каждой из них в виде БИС. Все они образуют комплект секционных интегральных схем МП.

Таким образом, **микропроцессорная секция** – это БИС, предназначенная для обработки нескольких разрядов данных или выполнения определенных управляющих операций. Секционность интегральных схем МП определяет возможность "наращивания" разрядности обрабатываемых данных или усложнения устройств управления микропроцессора при "параллельном" включении большего числа БИС.

Однокристалльные и трехкристалльные интегральных схем МП, как правило, изготавливают на основе микроэлектронных технологий униполярных полупроводниковых приборов, а многокристалльные секционные БИС МП на основе технологии биполярных полупроводниковых приборов. Использование многокристалльных микропроцессорных высокоскоростных биполярных БИС, имеющих функциональную законченность при малой физической разрядности обрабатываемых данных и монтируемых в корпус с большим числом выводов, позволяет организовать разветвление связи в процессоре, а также осуществить конвейерные принципы обработки информации для повышения его производительности.

2. По назначению различают *универсальные и специализированные* микропроцессоры [4].

Универсальные микропроцессоры могут быть применены для решения широкого круга разнообразных задач. При этом их эффективная производительность слабо зависит от проблемной специфики решаемых задач. Специализация МП, т. е. его проблемная ориентация на ускоренное выполнение определенных функций позволяет резко увеличить эффективную производительность при решении только определенных задач.

Среди специализированных микропроцессоров можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей логических операций, математические МП, предназначенные для повышения производительности при выполнении арифметических операций за счет, например, матричных методов их выполнения, МП для обработки данных в различных областях приме-

нений и т. д. С помощью специализированных МП можно эффективно решать новые сложные задачи параллельной обработки данных. Например, конволюция позволяет осуществить более сложную математическую обработку сигналов, чем широко используемые методы корреляции. Последние в основном сводятся к сравнению всего двух серий данных: входных, передаваемых формой сигнала, и фиксированных опорных, а также к определению их подобия. Конволюция дает возможность в реальном масштабе времени находить соответствие для сигналов изменяющейся формы путем сравнения их с различными эталонными сигналами, что, например, может позволить эффективно выделить полезный сигнал на фоне шума. Разработанные однокристалльные конвольверы используются в устройствах опознавания образов в тех случаях, когда возможности сбора данных превосходят способности системы обрабатывать эти данные.

3. По виду обрабатываемых входных сигналов различают *цифровые* и *аналоговые* микропроцессоры [3, 4].

Сами микропроцессоры – цифровые устройства, однако, могут иметь встроенные аналого-цифровые и цифроаналоговые преобразователи. Поэтому входные аналоговые сигналы передаются в МП через преобразователь в цифровой форме, обрабатываются и после обратного преобразования в аналоговую форму поступают на выход. С архитектурной точки зрения такие микропроцессоры представляют собой аналоговые функциональные преобразователи сигналов и называются **аналоговыми микропроцессорами**. Они выполняют функции любой аналоговой схемы (например, производят генерацию колебаний, модуляцию, смещение, фильтрацию, кодирование и декодирование сигналов в реальном масштабе времени и т. д., заменяя сложные схемы, состоящие из операционных усилителей, катушек индуктивности, конденсаторов и т. д.). При этом применение аналогового микропроцессора значительно повышает точность обработки аналоговых сигналов и их воспроизводимость, а также расширяет функциональные возможности за счет программной "настройки" цифровой части микропроцессора на различные алгоритмы обработки сигналов.

Обычно в составе однокристалльных аналоговых МП имеется несколько каналов аналого-цифрового и цифроаналогового преобразования. В аналоговом микропроцессоре разрядность обрабатываемых данных достигает 24 бит и более, большое значение уделяется увеличению скорости выполнения арифметических операций.

Отличительная черта аналоговых микропроцессоров – способность к переработке большого объема числовых данных, т. е. к выполнению операций сложения и умножения с большой скоростью при не-

обходимости даже за счет отказа от операций прерываний и переходов. Аналоговый сигнал, преобразованный в цифровую форму, обрабатывается в реальном масштабе времени и передается на выход обычно в аналоговой форме через цифроаналоговый преобразователь. При этом, согласно теореме Котельникова, частота квантования аналогового сигнала должна вдвое превышать верхнюю частоту сигнала.

Сравнение цифровых микропроцессоров производится сопоставлением времени выполнения ими списков операций. Сравнение же аналоговых микропроцессоров производится по количеству эквивалентных звеньев аналого-цифровых фильтров рекурсивных фильтров второго порядка. Производительность аналогового микропроцессора определяется его способностью быстро выполнять операции умножения: чем быстрее осуществляется умножение, тем больше эквивалентное количество звеньев фильтра в аналоговом преобразователе и тем более сложный алгоритм преобразования цифровых сигналов можно задавать в микропроцессоре.

Одним из направлений дальнейшего совершенствования аналоговых микропроцессоров является повышение их универсальности и гибкости. Поэтому вместе с повышением скорости обработки большого объема цифровых данных будут развиваться средства обеспечения развитых вычислительных процессов обработки цифровой информации за счет реализации аппаратных блоков прерывания программ и программных переходов.

4. По характеру временной организации работы микропроцессоры делят на *синхронные* и *асинхронные* [3, 4].

Синхронные микропроцессоры – микропроцессоры, в которых начало и конец выполнения операций задаются устройством управления (время выполнения операций в этом случае не зависит от вида выполняемых команд и величин операндов).

Асинхронные микропроцессоры позволяют начало выполнения каждой следующей операции определить по сигналу фактического окончания выполнения предыдущей операции. Для более эффективного использования каждого устройства микропроцессорной системы в состав асинхронно работающих устройств вводят электронные цепи, обеспечивающие автономное функционирование устройств. Закончив работу над какой-либо операцией, устройство вырабатывает сигнал запроса, означающий его готовность к выполнению следующей операции. При этом роль естественного распределителя работ принимает на себя память, которая в соответствии с заранее установленным приоритетом выполняет запросы остальных устройств по обеспечению их командной информацией и данными.

5. По организации структуры микропроцессорных систем различают одно- и многомагистральные микроЭВМ [5].

В одномагистральных микроЭВМ все устройства имеют одинаковый интерфейс и подключены к единой информационной магистрали, по которой передаются коды данных, адресов и управляющих сигналов.

В многомагистральных микроЭВМ устройства группами подключаются к своей информационной магистрали. Это позволяет осуществить одновременную передачу информационных сигналов по нескольким (или всем) магистралям. Такая организация систем усложняет их конструкцию, однако увеличивает производительность.

6. По количеству выполняемых программ различают одно- и многопрограммные микропроцессоры.

В однопрограммных микропроцессорах выполняется только одна программа. Переход к выполнению другой программы происходит после завершения текущей программы.

В много- или мультипрограммных микропроцессорах одновременно выполняется несколько (обычно несколько десятков) программ. Организация мультипрограммной работы микропроцессорных управляющих систем позволяет осуществить контроль за состоянием и управлением большого числа источников или приемников информации.

3. Архитектура микропроцессора

3.1. Понятие архитектуры микропроцессора

С точки зрения пользователя (разработчика автоматических систем) при выборе МП для решения конкретной задачи целесообразно располагать некоторыми обобщенными или комплексными характеристиками возможностей МП, т. е. воспринимать его как нечто цельное, имеющее вполне определенные потребительские качества (свойства и характеристики). В конечном итоге разработчик нуждается в уяснении и понимании лишь тех компонентов МП и МПС, которые явно отражаются в программах и (или) должны быть учтены при разработке и выполнении программ: число и имена программно-доступных регистров; разрядность машинного слова; система команд; доступный размер и адреса ОЗУ; быстродействие МП; схему обработки прерываний; способы адресации ОЗУ и внешних устройств. Совокупность таких сведений представляет определенную модель МП (МПС) с точки зрения пользователя (разработчика МПАС).

Указанные выше характеристики и свойства определяются понятием архитектуры МП (МПС, МЭВМ).

Архитектура МП – это его логическая организация, рассматриваемая с точки зрения пользователя; она определяет возможности МП по аппаратной, программной и микропрограммной реализации функций, необходимых для построения МПС и МПАС [2, 3].

Понятие архитектуры МП отражает:

- структуру, т. е. совокупность компонентов, составляющих МП, и связей между ними;
- способы представления и форматы данных;
- способы обращения ко всем доступным для пользователя (программно-доступным) элементам структуры (адресация к регистрам, ячейкам оперативной и постоянной памяти, внешним устройствам);
- набор операций, выполняемых МП, т. е. система команд МП;
- характеристики управляющих слов и сигналов, вырабатываемых микропроцессором и поступающих в МП извне;
- реакцию на внешние сигналы (схема обработки прерываний и т. д.) и другие характеристики. Ниже будут рассмотрены некоторые элементы архитектуры.

3.2. Основные характеристики микропроцессора

Микропроцессор характеризуется:

- 1) тактовой частотой, определяющей максимальное время выполнения переключения элементов в ЭВМ;
- 2) разрядностью, т. е. максимальным числом одновременно обрабатываемых двоичных разрядов;
- 3) архитектурой.

Разрядность МП обозначается $m/n/k$ и включает:

m – разрядность внутренних регистров, определяет принадлежность к тому или иному классу процессоров;

n – разрядность шины данных, определяет скорость передачи информации;

k – разрядность шины адреса, определяет размер адресного пространства.

Например, МП i8088 характеризуется значениями $m/n/k = 16/8/20$;

Понятие архитектуры микропроцессора включает в себя систему команд и способы адресации, возможность совмещения выполнения команд во времени, наличие дополнительных устройств в составе микропроцессора, принципы и режимы его работы. Выделяют понятия микроархитектуры и макроархитектуры.

Микроархитектура микропроцессора – это аппаратная организация и логическая структура микропроцессора, регистры, управляющие схемы, арифметико-логические устройства, запоминающие устройства и связывающие их информационные магистрали [1, 2, 3].

Макроархитектура – это система команд, типы обрабатываемых данных, режимы адресации и принципы работы микропроцессора.

В общем случае под архитектурой ЭВМ понимается абстрактное представление машины в терминах основных функциональных модулей, языка ЭВМ, структуры данных [1, 2, 3].

3.3. Типы архитектур микропроцессоров

Все микропроцессоры можно разделить на следующие группы:

- МП с гарвардской архитектурой;
- МП с фоннеймановской архитектурой;
- МП типа CISC (Complex Instruction Set Computing) с полным набором команд;
- МП типа RISC (Reduced Instruction Set Computing) с сокращенным набором команд;
- МП типа MISC (Minimum Instruction Set Computing) с минимальным набором команд и весьма высоким быстродействием (в настоящее время эти модели находятся в стадии разработки).

3.3.1. Гарвардская и фоннеймановская архитектуры микропроцессора

Архитектуры микропроцессоров различаются по использованию памяти. Наибольшее распространение получили:

- гарвардская архитектура;
- архитектура фон Неймана.

Гарвардская архитектура предполагает раздельное использование памяти программ и данных. Обычно такую архитектуру используют для повышения быстродействия системы за счёт разделения путей доступа к памяти программ и данных. Большинство специализированных микропроцессоров (особенно микроконтроллеры) имеют данную архитектуру.

Антипод гарвардской – архитектура фон Неймана – предполагает хранение программ и данных в общей памяти и наиболее характерна для микропроцессоров, ориентированных на использование в компьютерах. Примером могут служить микропроцессоры семейства x86.

3.3.2. Микропроцессоры типа CISC

Термин *CISC* означает сложную систему команд и является аббревиатурой английского определения Complex Instruction Set Computer.

Большинство современных ПК типа IBM PC (International Business Machine) используют МП типа CISC, характеристики наиболее распространенных из них приведены в табл. 1.

Таблица 1

Характеристики наиболее распространенных CISC МП

Мо- дель МП	Разрядность, бит		Так- товая частота, МГц	Адресное пространство, байт	Число ко- манд	Число элемен- тов	Год выпус- ка
	данных	адреса					
4004	4	4	4,77	4*103	45	2300	1971
8080	8	8	4,77	64*103		10000	1974
8086	16	16	4,77 и 8	106"	134	70000	1982
8088	8, 16	16	4,77 и 8	106	134	70000	1981
80186	16	20	8 и 10	106		140000	1984
80286	16	24	10–33	4*106 (вирту- альное 109)		180000	1985
80386	32	32	25–50	16*106 (вир- туальное 4*109)	240	275000	1987
80486	32	32	33–100	16*106 (вир- туальное 4*109)	240	1,2x106	1989
Pentium	64	32	50–150	4*109	240	3,1*106	1993
Pentium Pro	64	32	66–200	4*109	240	5,5*106	1995

Отметим некоторые характеристики МП:

- начиная с МП 80386 используется конвейерное выполнение команд – одновременное выполнение разных тактов последовательных команд в разных частях МП при непосредственной передаче результатов из одной части МП в другую. Конвейерное выполнение команд увеличивает эффективное быстродействие ПК в 2–3 раза;
- начиная с МП 80286 предусматривается возможность работы в вычислительной сети;
- начиная с МП 80286 имеется возможность многозадачной работы (многопрограммность) и сопутствующая ей защита памяти;
- начиная с МП 80386 обеспечивается поддержка режима системы виртуальных машин, т. е. такого режима многозадачной работы, при котором в одном МП моделируется как бы несколько компьютеров, работающих параллельно и имеющих разные операционные системы;
- начиная с МП 80286 микропроцессоры могут работать в двух режимах: реальном (Real mode) и защищенном (Protected mode). В ре-

альном режиме имитируется (эмулируется) работа МП 8086, естественно, однозадачная. В защищенном режиме возможна многозадачная работа с непосредственным доступом к расширенной памяти (см. подразд. 4.5) и с защитой памяти, отведенной задачам, от посторонних обращений.

Микропроцессоры 80586 (P5) более известны по их товарной марке Pentium, которая запатентована фирмой Intel (МП 80586 других фирм имеют иные обозначения: K5 у фирмы AMD, M 1 у фирмы Cyrix и др.).

Эти микропроцессоры имеют пятиступенную конвейерную структуру, обеспечивающую многократное совмещение тактов выполнения последовательных команд, и КЭШ-буфер для команд условной передачи управления, позволяющий предсказывать направление ветвления программ; по эффективному быстродействию они приближаются к RISC МП, выполняющим каждую команду как бы за один такт. Pentium имеют 32-разрядную адресную шину и 64-разрядную шину данных. Обмен данными с системой может выполняться со скоростью 1 Гб/с.

У всех МП Pentium имеется встроенная КЭШ-память, отдельно для команд, отдельно для данных; имеются специализированные конвейерные аппаратные блоки сложения, умножения и деления, значительно ускоряющие выполнение операций с плавающей запятой.

Микропроцессоры Pentium Pro. В сентябре 1995 г. прошли презентацию и выпущены МП 80686 (P6), торговая марка Pentium Pro. Благодаря новым схемотехническим решениям они обеспечивают для ПК более высокую производительность. Часть этих новшеств может быть объединена понятием динамическое исполнение (dynamic execution), что в первую очередь означает наличие 14-ступенной суперконвейерной структуры (superpipelining), предсказания ветвлений программы при условных передачах управления (branch prediction) и исполнение команд по предполагаемому пути ветвления (speculative execution).

КЭШ-память емкостью 256–512 кб – обязательный атрибут высокопроизводительных систем на процессорах Pentium. Однако у них встроенная КЭШ-память имеет небольшую емкость (16 кб), а основная ее часть находится вне процессора на материнской плате. Поэтому обмен данными с ней происходит не на внутренней частоте МП, а на частоте тактового генератора, которая обычно в 2–3 раза ниже, что снижает общее быстродействие компьютера. В МП Pentium Pro КЭШ-память емкостью 256–512 кб находится в самом микропроцессоре.

Микропроцессоры OverDrive. Интерес представляют также недавно разработанные МП OverDrive, по существу являющиеся своеобраз-

ными сопроцессорами, обеспечивающими для МП 80486 режимы работы и эффективное быстроедействие, характерные для МП Pentium. Появились МП OverDrive, улучшающие характеристики и микропроцессоров Pentium.

3.3.3. Микропроцессоры типа RISC

Термин *RISC* означает сокращённую систему команд и происходит от английского Reduced Instruction Set Computer.

Микропроцессоры типа RISC содержат набор только простых, чаще всего встречающихся в программах команд. При необходимости выполнения более сложных команд в микропроцессоре производится их автоматическая сборка из простых. В этих МП на выполнение каждой простой команды за счет их наложения и параллельного выполнения тратится 1 машинный такт (на выполнение даже самой короткой команды из системы CISC обычно тратится 4 такта).

Некоторые микропроцессоры типа RISC: ARM (на его основе выпускались ПК IBM PC RT) – один из первых 32-разрядных RISC микропроцессоров, имеющий 118 различных команд. Современные RISC микропроцессоры (80860, 80960, 80870, Power PC) являются 64-разрядными при быстроедействии до 150 млн оп./с. Микропроцессоры Power PC (Performance Optimized With Enhanced RISC PC) весьма перспективны и уже сейчас широко применяются в машинах-серверах и в ПК типа Macintosh.

Микропроцессоры типа RISC имеют очень высокое быстроедействие, но программно не совместимы с CISC-процессорами: при выполнении программ, разработанных для ПК типа IBM PC, они могут лишь эмулировать (моделировать, имитировать) МП типа CISC на программном уровне, что приводит к резкому уменьшению их эффективной производительности.

Однако, несмотря на широкую распространённость этих понятий, необходимо признать, что сами названия не отражают главного различия между системами команд CISC и RISC. *Основная идея RISC-архитектуры* – это тщательный подбор таких комбинаций кодов операций, которые можно было бы выполнить за один такт тактового генератора. Основной выигрыш от такого подхода – резкое упрощение аппаратной реализации ЦП и возможность значительно повысить его производительность. Все команды работают с операндами и имеют одинаковый формат. Обращение к памяти выполняется с помощью специальных команд загрузки регистра и записи. Простота структуры и небольшой набор команд позволяет реализовать полностью их аппа-

ратное выполнение и эффективный конвейер при небольшом объеме оборудования. Арифметику RISC-процессоров отличает высокая степень дробления конвейера. Этот прием позволяет увеличить тактовую частоту (значит и производительность) компьютера. Чем более элементарные действия выполняются в каждой фазе работы конвейера, тем выше частота его работы. RISC-процессоры с самого начала ориентированы на реализацию всех возможностей ускорения арифметических операций, поэтому их конвейеры обладают значительно более высоким быстродействием, чем в CISC-процессорах. Поэтому RISC-процессоры в 2–4 раза быстрее имеющих ту же тактовую частоту CISC-процессоров с обычной системой команд и высокопроизводительней, несмотря на больший объем программ, на (30 %). Дейв Паттерсон и Карло Секуин сформулировали 4 основных принципа RISC:

1. Любая операция должна выполняться за один такт, вне зависимости от ее типа.

2. Система команд должна содержать минимальное количество наиболее часто используемых простейших инструкций одинаковой длины.

3. Операции обработки данных реализуются только в формате “регистр–регистр” (операнды выбираются из оперативных регистров процессора, и результат операции записывается также в регистр; а обмен между оперативными регистрами и памятью выполняется только с помощью команд чтения/записи).

4. Состав системы команд должен быть “удобен” для компиляции операторов языков высокого уровня.

Микропроцессоры с архитектурой CISC (Complex Instruction Set Computers) – архитектура вычислений с полной системой команд. Реализующие на уровне машинного языка комплексные наборы команд различной сложности (от простых, характерных для микропроцессора первого поколения, до команд значительной сложности, характерных для современных 32-разрядных микропроцессоров типа 80486, 68040 и др.).

3.4. Архитектурно-функциональные принципы построения ЭВМ

Основополагающие принципы цифровой ЭВМ (Джон фон Нейман, 1946 г.):

1. Принцип использования двоичной системы исчисления для представления информации в ЭВМ.
2. Принцип программного управления ЭВМ.
3. Принцип условного перехода.
4. Принцип хранимой программы.
5. Принцип иерархичности запоминающих устройств ЭВМ.

В цифровых вычислительных устройствах все операции (математические и логические) производятся над числами (величинами), представленными в кодовом выражении. Причём в данном случае все операции производятся в одном и том же узле. Большинство задач цифровой обработки может быть решено при помощи устройств, математическая структура которых изображена на рис. 2.

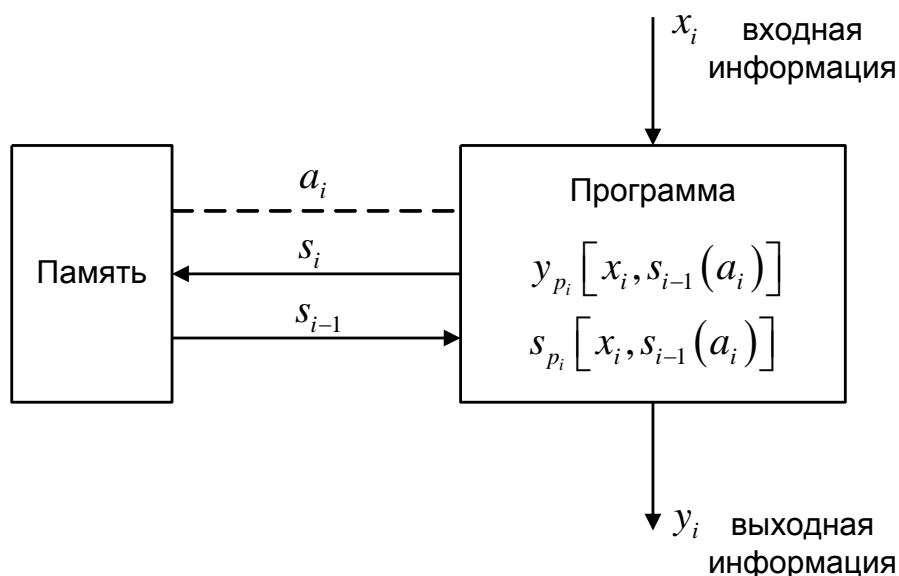


Рис. 2. Схема конечного автомата фон Неймана

На вход данного устройства (рис. 2) подаётся массив входной информации

$$x_i = \{a_i, p_i, x_i^{(0)}, x_i^{(1)}, \dots, x_i^{(n)}\}, \quad (1)$$

где a_i – адресная часть массива, указывающая ячейку (ячейки) памяти, содержание которой $s_{i-1}(a_i)$ необходимо для обработки информации; p_i – номер программы (адрес программы), по которой должна быть произведена обработка; $x_i^{(0)}, \dots, x_i^{(n)}$ – множество входных величин, наиболее полно и точно отражающее входную информацию величины x .

В ходе обработки получается выходной массив

$$y_i = y_{p_i} [x_i, s_{i-1}(a_i)], \quad (2)$$

где y_i – выходной массив, который образуется путём обработки x_i программой p_i по адресу $s_{i-1}(a_i)$.

Выходной массив y_i представляется через множество выходных элементов

$$y_i = (y_i^{(0)}, \dots, y_i^{(n)}). \quad (3)$$

В память ЭВМ помещается массив

$$s_i = s_{p_i} [x_i, s_{i-1}(a_i)], \quad (4)$$

т. е. новый массив памяти.

3.5. Структура типовой ЭВМ (персонального компьютера)

3.5.1. Основные блоки ПК и их назначение

Архитектура компьютера обычно определяется совокупностью ее свойств, существенных для пользователя. Основное внимание при этом уделяется структуре и функциональным возможностям машины, которые можно разделить на основные и дополнительные.

Основные функции определяют назначение ЭВМ: обработка и хранение информации, обмен информацией с внешними объектами. *Дополнительные* функции повышают эффективность выполнения основных функций: обеспечивают эффективные режимы ее работы, диалог с пользователем, высокую надежность и др. Названные функции ЭВМ реализуются с помощью ее компонентов: аппаратных и программных средств.

Структура компьютера – это некоторая модель, устанавливающая состав, порядок и принципы взаимодействия входящих в нее компонентов.

Персональный компьютер (ПК) – это настольная или переносная ЭВМ, удовлетворяющая требованиям общедоступности и универсальности применения.

Достоинствами ПК являются:

- малая стоимость, находящаяся в пределах доступности для индивидуального покупателя;
- автономность эксплуатации без специальных требований к условиям окружающей среды;
- гибкость архитектуры, обеспечивающая ее адаптивность к разнообразным применениям в сфере управления, науки, образования, в быту;
- "дружественность" операционной системы и прочего программного обеспечения, обуславливающая возможность работы с ней пользователя без специальной профессиональной подготовки;
- высокая надежность работы (более 5 тыс. ч наработки на отказ).

Структура типовой ЭВМ представлена на рис. 3.

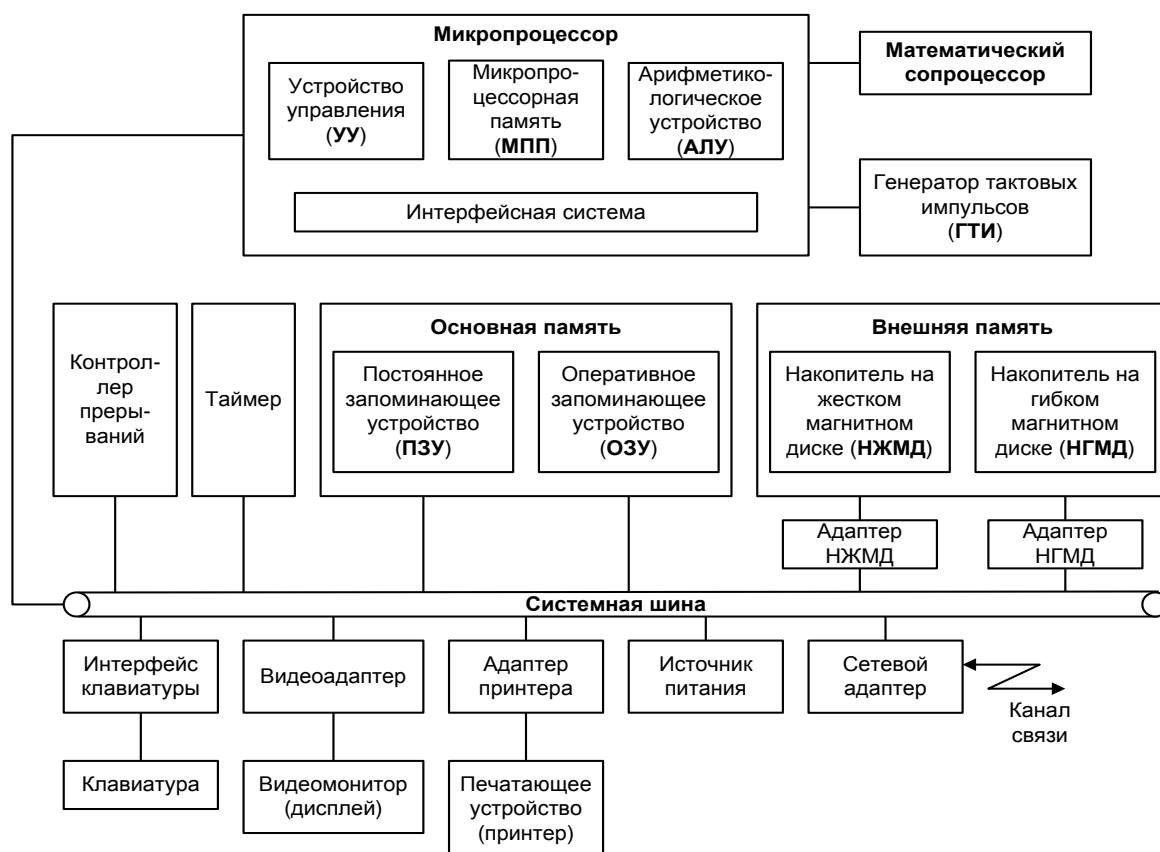


Рис. 3. Структура типовой ЭВМ

Микропроцессор (МП) – это центральный блок ПК, предназначенный для управления работой всех блоков машины и для выполнения арифметических и логических операции над информацией.

В состав микропроцессора входят:

- *устройство управления (УУ)* – формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы), обусловленные спецификой выполняемой операции и результатами предыдущих операций; формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки ЭВМ. Опорную последовательность импульсов устройство управления получает от генератора тактовых импульсов;

- *арифметико-логическое устройство (АЛУ)* – предназначено для выполнения всех арифметических и логических операций над числовой и символьной информацией (в некоторых моделях ПК для ускорения выполнения операций к АЛУ подключается дополнительный математический сопроцессор);

- *микропроцессорная память (МПП)* – служит для кратковременного хранения, записи и выдачи информации, непосредственно используемой в вычислениях в ближайшие такты работы машины. МПП строится на регистрах и используется для обеспечения высокого быстродействия машины, т. к. основная память (ОП) не всегда обеспечивает скорость записи, поиска и считывания информации, необходимую для эффективной работы быстродействующего микропроцессора. **Регистры** – быстродействующие ячейки памяти различной длины (в отличие от ячеек ОП, имеющих стандартную длину 1 байт и более низкое быстродействие);

- *интерфейсная система микропроцессора* – реализует сопряжение и связь с другими устройствами ПК; включает в себя внутренний интерфейс МП, буферные запоминающие регистры и схемы управления портами ввода/вывода (ПВВ) и системной шиной. **Интерфейс (interface)** – совокупность средств сопряжения и связи устройств компьютера, обеспечивающая их эффективное взаимодействие. **Порт ввода/вывода (I/O \approx Input/Output port)** – аппаратура сопряжения, позволяющая подключить к микропроцессору другое устройство ПК.

Генератор тактовых импульсов генерирует последовательность электрических импульсов; частота генерируемых импульсов определяет тактовую частоту машины. Промежуток времени между соседними импульсами определяет время одного такта работы машины или просто такт работы машины.

Частота генератора тактовых импульсов является одной из основных характеристик персонального компьютера и во многом определяет скорость его работы, ибо каждая операция в машине выполняется за определенное количество тактов.

Системная шина – это основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой.

Системная шина включает в себя:

- *кодovou шину данных (КШД)*, содержащую провода и схемы сопряжения для параллельной передачи всех разрядов числового кода (машинного слова) операнда;
- *кодovou шину адреса (КША)*, включающую провода и схемы сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода/вывода внешнего устройства;
- *кодovou шину инструкций (КШИ)*, содержащую провода и схемы сопряжения для передачи инструкций (управляющих сигналов, импульсов) во все блоки машины;
- *шину питания*, имеющую провода и схемы сопряжения для подключения блоков ПК к системе энергопитания.

Системная шина обеспечивает три направления передачи информации:

- 1) между микропроцессором и основной памятью;
- 2) между микропроцессором и портами ввода/вывода внешних устройств;
- 3) между основной памятью и портами ввода/вывода внешних устройств (в режиме прямого доступа к памяти).

Все блоки, а точнее их порты ввода/вывода, через соответствующие унифицированные разъемы (стыки) подключаются к шине единообразно: непосредственно или через *контроллеры (адаптеры)*. Управление системной шиной осуществляется микропроцессором либо непосредственно, либо, что чаще, через дополнительную микросхему – *контроллер шины*, формирующей основные сигналы управления.

Основная память предназначена для хранения и оперативного обмена информацией с прочими блоками машины. ОП содержит два вида запоминающих устройств: постоянное запоминающее устройство (ПЗУ) и оперативное запоминающее устройство (ОЗУ).

ПЗУ служит для хранения неизменяемой (постоянной) программной и справочной информации, позволяет оперативно только считывать хранящуюся в нем информацию (изменить информацию в ПЗУ нельзя).

ОЗУ предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе, выполняемом ПК в текущий период времени. Главными достоинствами оперативной памяти являются ее высокое быстродействие и возможность обращения к каж-

дой ячейке памяти отдельно (прямой адресный доступ к ячейке), В качестве недостатка ОЗУ следует отметить невозможность сохранения информации в ней после выключения питания машины (энергозависимость).

Внешняя память относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач. В частности, во внешней памяти хранится все программное обеспечение компьютера. Внешняя память содержит разнообразные виды запоминающих устройств, но наиболее распространенными, имеющимися практически на любом компьютере, являются накопители на жестких (НЖМД) и гибких (НГМД) магнитных дисках.

Назначение этих накопителей – хранение больших объемов информации, запись и выдача хранимой информации по запросу в оперативное запоминающее устройство. Различаются НЖМД и НГМД лишь конструктивно, объемами хранимой информации и временем поиска, записи и считывания информации.

В качестве устройств внешней памяти используются также запоминающие устройства на кассетной магнитной ленте (стримеры), накопители на оптических дисках (CD-ROM – Compact Disk Read Only Memory – компакт-диск с памятью, только читаемой) и др.

Источник питания – это блок, содержащий системы автономного и сетевого энергопитания ПК.

Таймер – это внутримашинные электронные часы, обеспечивающие при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд). Таймер подключается к автономному источнику питания – аккумулятору – и при отключении машины от сети продолжает работать.

Внешние устройства – это важнейшая составная часть любого вычислительного комплекса. Достаточно сказать, что по стоимости ВУ иногда составляют 50–80 % стоимости всего ПК. От состава и характеристик ВУ во многом зависят возможность и эффективность применения ПК в системах управления и в народном хозяйстве в целом.

Внешние устройства ПК обеспечивают взаимодействие машины с окружающей средой; пользователями, объектами управления и другими ЭВМ. ВУ весьма разнообразны и могут быть классифицированы по ряду признаков. Так, по назначению можно выделить следующие виды ВУ:

- внешние запоминающие устройства (ВЗУ) или внешняя память ПК;
- диалоговые средства пользователя;

- устройства ввода информации;
- устройства вывода информации;
- средства связи и телекоммуникации.

Диалоговые средства пользователя включают в свой состав видеомониторы (дисплеи), реже – пультовые пишущие машинки (принтеры с клавиатурой) и устройства речевого ввода/вывода информации.

Видеомонитор (дисплей) – устройство для отображения вводимой и выводимой из ПК информации.

Устройства речевого ввода/вывода относятся к быстроразвивающимся средствам мультимедиа. Устройства речевого ввода – это различные микрофонные акустические системы, "звуковые мыши", например, со сложным программным обеспечением, позволяющим распознавать произносимые человеком буквы и слова, идентифицировать их и кодировать.

Устройства речевого вывода – это различные синтезаторы звука, выполняющие преобразование цифровых кодов в буквы и слова, воспроизводимые через громкоговорители (динамики) или звуковые колонки, подсоединенные к компьютеру.

К устройствам ввода информации относятся:

- *клавиатура* – устройство для ручного ввода числовой, текстовой и управляющей информации в ПК;
- *графические планшеты (диджитайзеры)* – для ручного ввода графической информации, изображений путем перемещения по планшету специального указателя (пера); при перемещении пера автоматически выполняются считывание координат его местоположения и ввод этих координат в ПК;
- *сканеры* – для автоматического считывания с бумажных носителей и ввода в ПК машинописных текстов, графиков, рисунков, чертежей; в устройстве кодирования сканера в текстовом режиме считанные символы после сравнения с эталонными контурами специальными программами преобразуются в коды ASCII, а в графическом режиме считанные графики и чертежи преобразуются в последовательности двумерных координат;
- *манипуляторы (устройства указания)*: джойстик – рычаг, мышь, трекбол – шар в оправе, световое перо и др. – для ввода графической информации на экран дисплея путем управления движением курсора по экрану с последующим кодированием координат курсора и вводом их в ПК;
- *сенсорные экраны* – для ввода отдельных элементов изображения, программ или команд с полиэкрана дисплея в ПК.

К устройствам вывода информации относятся:

- *принтеры* – печатающие устройства для регистрации информации на бумажный носитель (см. подразд. 4.5);
- *графопостроители (плоттеры)* – для вывода графической информации (графиков, чертежей, рисунков) из ПК на бумажный носитель; плоттеры бывают векторные с вычерчиванием изображения с помощью пера и растровые: термографические, электростатические, струйные и лазерные. По конструкции плоттеры подразделяются на планшетные и барабанные. Основные характеристики всех плоттеров примерно одинаковые: скорость вычерчивания – 100–1000 мм/с, у лучших моделей возможно цветное изображение и передача полутонов; наибольшая разрешающая способность и четкость изображения – у лазерных плоттеров, но они самые дорогие.

Устройства связи и телекоммуникации используются для связи с приборами и другими средствами автоматизации (согласователи интерфейсов, адаптеры, цифроаналоговые и аналого-цифровые преобразователи и т. п.) и для подключения ПК к каналам связи, к другим ЭВМ и вычислительным сетям (сетевые интерфейсные платы, "стыки", мультиплексоры передачи данных, модемы).

В частности, показанный на рис. 3 сетевой адаптер является внешним интерфейсом ПК и служит для подключения его к каналу связи для обмена информацией с другими ЭВМ, для работы в составе вычислительной сети. В глобальных сетях функции сетевого адаптера выполняет модулятор-демодулятор.

Многие из названных выше устройств относятся к условно выделенной группе – средствам мультимедиа.

Средства мультимедиа (multimedia – многосредовость) – это комплекс аппаратных и программных средств, позволяющих человеку общаться с компьютером, используя самые разные естественные для себя среды: звук, видео, графику, тексты, анимацию и др.

К средствам мультимедиа относятся устройства речевого ввода и вывода информации; широко распространенные уже сейчас сканеры (поскольку они позволяют автоматически вводить в компьютер печатные тексты и рисунки); высококачественные видео- (video-) и звуковые (sound-) платы, платы видеозахвата (videograbber), снимающие изображение с видеомagneтофона или видеокамеры и вводящие его в ПК; высококачественные акустические и видеовоспроизводящие системы с усилителями, звуковыми колонками, большими видеоэкранами. Но, пожалуй, еще с большим основанием к средствам мультимедиа относят внешние запоминающие устройства большой емкости на оптических дисках, часто используемые для записи звуковой и видеоинформации.

Стоимость компактных дисков (CD) при их массовом тиражировании невысокая, а учитывая их большую емкость (650 Мб, а новых типов – 1 Гб и выше), высокую надежность и долговечность, стоимость хранения информации на CD для пользователя оказывается несравнимо меньшей, нежели на магнитных дисках. Это уже привело к тому, что большинство программных средств самого разного назначения поставляется на CD. На компакт-дисках за рубежом организуются обширные базы данных, целые библиотеки; на CD представлены словари, справочники, энциклопедии; обучающие и развивающие программы по общеобразовательным и специальным предметам.

CD широко используются, например, при изучении иностранных языков, правил дорожного движения, бухгалтерского учета, законодательства вообще и налогового законодательства в частности. И все это сопровождается текстами и рисунками, речевой информацией и мультипликацией, музыкой и видео. В чисто бытовом аспекте CD можно использовать для хранения аудио- и видеозаписей, т. е. использовать вместо плеерных аудиокассет и видеокассет. Следует упомянуть, конечно, и о большом количестве программ, компьютерных игр, хранимых на CD.

Таким образом, CD-ROM открывает доступ к огромным объемам разнообразной и по функциональному назначению, и по среде воспроизведения информации, записанной на компакт-дисках.

Дополнительные схемы. К системной шине и к микропроцессора ПК, наряду с типовыми внешними устройствами, могут быть подключены и некоторые дополнительные платы с интегральными микросхемами, расширяющие и улучшающие функциональные возможности микропроцессора: математический сопроцессор, контроллер прямого доступа к памяти, сопроцессор ввода/вывода, контроллер прерываний и др.

Математический сопроцессор широко используется для ускоренного выполнения операций над двоичными числами с плавающей запятой, над двоично-кодированными десятичными числами, для вычисления некоторых трансцендентных, в том числе тригонометрических, функций. Математический сопроцессор имеет свою систему команд и работает параллельно (совмещенно во времени) с основным МП, но под управлением последнего. Ускорение операций происходит в десятки раз. Последние модели МП, начиная с МП 80486 DX, включают сопроцессор в свою структуру.

Контроллер прямого доступа к памяти освобождает МП от прямого управления накопителями на магнитных дисках, что существенно повышает эффективное быстродействие ПК. Без этого контрол-

лера обмен данными между ВЗУ и ОЗУ осуществляется через регистр МП, а при его наличии данные непосредственно передаются между ВЗУ и ОЗУ, минуя МП.

Сопроцессор ввода/вывода за счет параллельной работы с МП значительно ускоряет выполнение процедур ввода/вывода при обслуживании нескольких внешних устройств (дисплей, принтер, НЖМД, НГМД и др.); освобождает МП от обработки процедур ввода/вывода, в том числе реализует и режим прямого доступа к памяти.

Важнейшую роль играет в ПК контроллер прерываний.

Прерывание – временный останов выполнения одной программы в целях оперативного выполнения другой, в данный момент более важной (приоритетной) программы.

Прерывания возникают при работе компьютера постоянно [4]. Достаточно сказать, что все процедуры ввода/вывода информации выполняются по прерываниям. Например, прерывания от таймера возникают и обслуживаются контроллером прерываний 18 раз в секунду (естественно, пользователь их не замечает).

Контроллер прерываний обслуживает процедуры прерывания, принимает запрос на прерывание от внешних устройств, определяет уровень приоритета этого запроса и выдает сигнал прерывания в МП. МП, получив этот сигнал, приостанавливает выполнение текущей программы и переходит к выполнению специальной программы обслуживания того прерывания, которое запросило внешнее устройство. После завершения программы обслуживания восстанавливается выполнение прерванной программы. Контроллер прерываний является программируемым.

Элементы конструкции ПК

Конструктивно ПК выполнены в виде центрального системного блока, к которому через разъемы подключаются внешние устройства: дополнительные устройства памяти, клавиатура, дисплей, принтер и др.

Системный блок обычно включает в себя *системную плату*, блок питания, накопители на дисках, разъемы для дополнительных устройств и *платы расширения* с контроллерами – адаптерами внешних устройств.

На системной плате (часто ее называют материнской платой Mother Board), как правило, размещаются :

- микропроцессор;
- математический сопроцессор;
- генератор тактовых импульсов;

- блоки (микросхемы) ОЗУ и ПЗУ;
- адаптеры клавиатуры, НЖМД и НГМД;
- контроллер прерываний;
- таймер и др.

3.5.2. Внутримашинный системный интерфейс

Внутримашинный системный интерфейс – система связи и сопряжения узлов и блоков ЭВМ между собой – представляет собой совокупность электрических линий связи (проводов), схем сопряжения с компонентами компьютера, протоколов (алгоритмов) передачи и преобразования сигналов.

Существуют два варианта организации внутримашинного интерфейса.

1. *Многосвязный интерфейс*: каждый блок ПК связан с прочими блоками своими локальными проводами; многосвязный интерфейс применяется, как правило, только в простейших бытовых ПК.

2. *Односвязный интерфейс*: все блоки ПК связаны друг с другом через общую или системную шину.

В подавляющем большинстве современных ПК в качестве системного интерфейса используется системная шина. Структура и состав системной шины были рассмотрены ранее. Важнейшими функциональными характеристиками системной шины являются: количество обслуживаемых ею устройств и ее пропускная способность, т. е. максимально возможная скорость передачи информации. Пропускная способность шины зависит от ее разрядности (есть шины 8-, 16-, 32- и 64-разрядные) и тактовой частоты, на которой шина работает.

В качестве системной шины в разных ПК использовались и могут использоваться:

- *шины расширений* – шины общего назначения, позволяющие подключать большое число самых разнообразных устройств,
- *локальные шины*, специализирующиеся на обслуживании небольшого количества устройств определенного класса.

Сравнительные технические характеристики некоторых шин приведены в табл. 2.

Шины расширений

• Шина *Multibus1* имеет две модификации: PC/XT bus (Personal Computer eXtended Technology – ПК с расширенной технологией) и PC/AT bus (PC Advanced Technology – ПК с усовершенствованной технологией).

- Шина *PC/XT bus* – 8-разрядная шина данных и 20-разрядная шина адреса, рассчитанная на тактовую частоту 4,77 МГц; имеет 4 линии для аппаратных прерываний и 4 канала для прямого доступа в память (каналы DMA – Direct Memory Access). Шина адреса ограничивала адресное пространство микропроцессора величиной 1 Мб. Используется с МП 8086, 8088.

- Шина *PC/AT bus* – 16-разрядная шина данных и 24-разрядная шина адреса, рабочая тактовая частота до 8 МГц, но может использоваться и МП с тактовой частотой 16 МГц, так как контроллер шины может делить частоту пополам; имеет 7 линий для аппаратных прерываний и 4 канала DMA. Используется с МП 80286.

- Шине *ISA* (Industry Standard Architecture – архитектура промышленного стандарта) – 16-разрядная шина данных и 24-разрядная шина адреса, рабочая тактовая частота 8 МГц, но может использоваться и МП с тактовой частотой выше 8 МГц (коэффициент деления увеличен); по сравнению с шинами *PC/XT* и *PC/AT* увеличено количество линий аппаратных прерываний с 7 до 15 и каналов прямого доступа к памяти DMA с 7 до 11. Благодаря 24-разрядной шине адреса, адресное пространство увеличилось с 1 до 16 Мб. Теоретическая пропускная способность шины данных равна 16 Мб/с, но реально она ниже (около 4-5 Мб/с), ввиду ряда особенностей ее использования. С появлением 32-разрядных высокоскоростных МП шина *ISA* стала существенным препятствием увеличения быстродействия ПК.

- Шина *EISA* (Extended ISA) – 32-разрядная шина данных и 32-разрядная шина адреса, создана в 1989 г. Адресное пространство шины 4 Гб, пропускная способность 33 Мб/с. Причем скорость обмена по каналу МП–КЭШ–ОП определяется параметрами микросхем памяти, увеличено число разъемов расширений (теоретически может подключаться до 15 устройств, практически – до 10). Улучшена система прерываний, шина *EISA* обеспечивает автоматическое конфигурирование системы и управление DMA; полностью совместима с шиной *ISA* (есть разъем для подключения *ISA*), шина поддерживает многопроцессорную архитектуру вычислительных систем. Шина *EISA* весьма дорогая и применяется в скоростных ПК, сетевых серверах и рабочих станциях.

- Шина *MCA* (Micro Channel Architecture) – 32-разрядная шина, созданная фирмой IBM в 1987 г. Для машин PS/2, пропускная способность 76 Мб/с, рабочая частота 10-20 МГц. По своим прочим характеристикам близка к шине *EISA*, но не совместима ни с *ISA*, ни с *EISA*. Поскольку ЭВМ PS/2 не получили широкого распространения, в первую очередь ввиду отсутствия наработанного обилия прикладных программ, шина *MCA* также используется не очень широко.

Локальные шины

Современные вычислительные системы характеризуются:

- стремительным ростом быстродействия микропроцессоров (например, МП Pentium может выдавать данные со скоростью 528 Мб/с по 64-разрядной шине данных) и некоторых внешних устройств. Так, для отображения цифрового полноэкранного видео с высоким качеством необходима пропускная способность 22 Мб/с;
- появлением программ, требующих выполнения большого количества интерфейсных операций (например, программы обработки графики в Windows, работа в среде Multimedia).

В этих условиях пропускной способности шин расширения, обслуживающих одновременно несколько устройств, оказалось недостаточно для комфортной работы пользователей, ибо компьютеры стали подолгу "задумываться".

Разработчики интерфейсов пошли по пути создания локальных шин, подключаемых непосредственно к шине МП и работающих на тактовой частоте МП (но не на внутренней рабочей его частоте), а также обеспечивающих связь с некоторыми скоростными внешними по отношению к МП устройствами: основной и внешней памятью, видеосистемами и др.

Сейчас существуют два основных стандарта универсальных локальных шин: VLB и PCI.

Шина VLB (VESA Local Bus – локальная шина VESA) – разработана в 1992 г. Ассоциацией стандартов видеооборудования (VESA – Video Electronics Standards Association), поэтому часто ее называют шиной VESA.

Шина VLB, по существу, является расширением внутренней шины МП для связи с видеоадаптером и реже с винчестером, платами Multimedia, сетевым адаптером. Разрядность шины – 32 бита, на подходе 64-разрядный вариант шины. Реальная скорость передачи данных по VLB – 80 Мб/с (теоретически достижимая – 132 Мб/с).

Недостатки шины:

- рассчитана на работу с МП 80386, 80486, пока не адаптирована для процессоров Pentium, Pentium Pro, Power PC;
- жесткая зависимость от тактовой частоты МП (каждая шина VLB рассчитана только на конкретную частоту);
- малое количество подключаемых устройств – к шине VLB могут подключаться только четыре устройства;
- отсутствует арбитраж шины – могут быть конфликты между подключаемыми устройствами.

Шина *PCI* (Peripheral Component Interconnect – соединение внешних устройств) – разработана в 1993 г. фирмой Intel.

Шина *PCI* является намного более универсальной, чем *VLB*, имеет свой адаптер, позволяющий ей настраиваться на работу с любым МП: 80486, Pentium, Pentium Pro, Celeron, Pentium 2 – Pentium 4, Power PC и др. Позволяет подключать 10 устройств самой разной конфигурации с возможностью автоконфигурирования, имеет свой "арбитраж", средства управления передачей данных.

Разрядность *PCI* – 32 бита с возможностью расширения до 64 бит, теоретическая пропускная способность 132 Мб/с, а в 64-битовом варианте – 263 Мб/с (реальная вдвое ниже).

Шина *PCI* хотя и является локальной, выполняет и многие функции шины расширения, в частности, шины расширения *ISA*, *EISA*, *MCA* (а она совместима с ними) при наличии шины *PCI* подключаются не непосредственно к МП (как это имеет место при использовании шины *VLB*), а к самой шине *PCI* (через интерфейс расширения).

Варианты конфигурации систем с шинами *VLB* и *PCI* показаны, соответственно, на рис. 4 и 5. Следует иметь в виду, что использование в ПК шин *VLB* и *PCI* возможно только при наличии соответствующей *VLB*- или *PCI*-материнской платы.

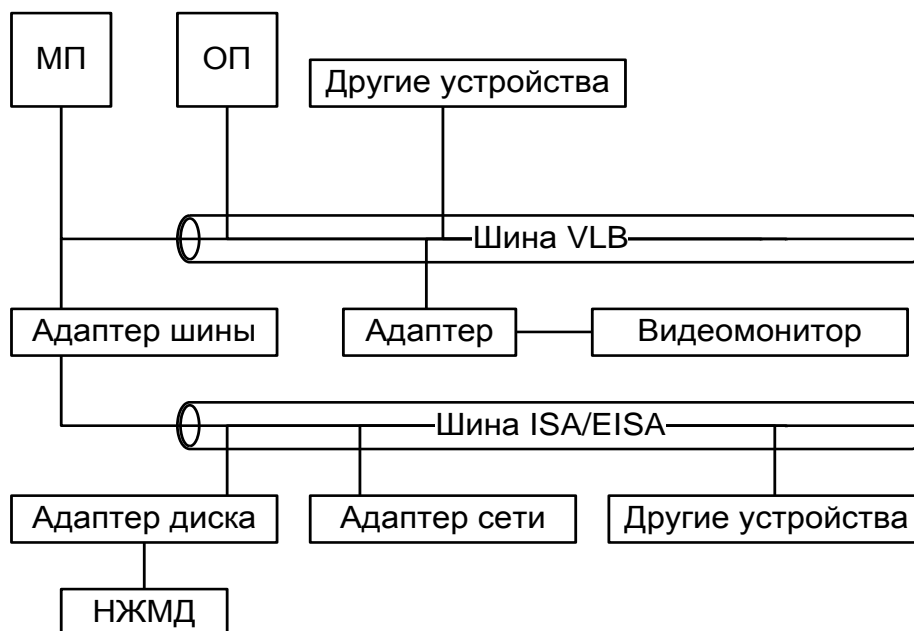


Рис. 4. Конфигурация системы с шиной *VLB*

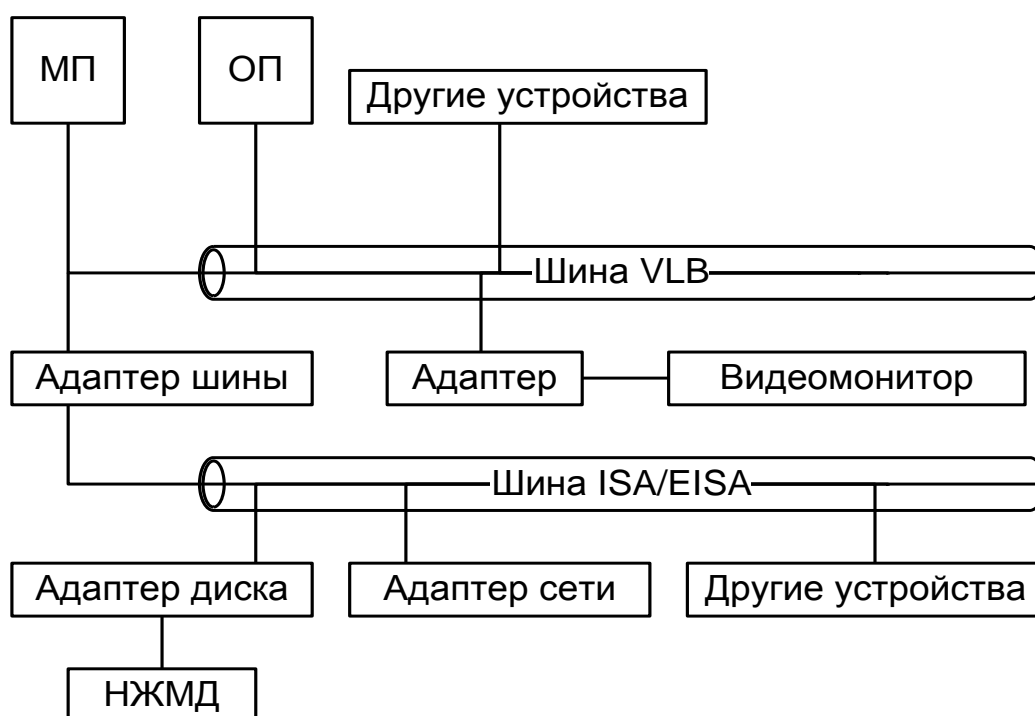


Рис. 5. Конфигурация системы с шиной PCI

Таблица 2

Основные характеристики шин

Параметр	ISA	EISA	MCA	VLB	PCI
Разрядность шины, бит	16	32	32;64	32;64	32; 64
Данных	24	32	32	32	32
Адреса					
Рабочая частота, МГц	8	8–33	10–20	до 33	до 33
Пропускная способность, Мб/с	4	33	76	132	132;264
теоретическая	2	8	20	80	50;100
практическая					
Число подключаемых устройств, шт.	6	15	15	4	10

3.5.3. Функциональные характеристики ПК

Основными характеристиками ПК являются:

1. Быстродействие, производительность, тактовая частота.

Единицами измерения быстродействия служат:

- МИПС (MIPS – Mega Instruction Per Second) – миллион операций над числами с фиксированной запятой (точкой);
- МФЛОПС (MFLOPS – Mega FLoating Operations Per Second) – миллион операций над числами с плавающей запятой (точкой);
- КОПС (KOPS – Kilo Operations Per Second) для низкопроизводительных ЭВМ – тысяча неких усредненных операций над числами;
- ГФЛОПС (GFLOPS – Giga FLoating Operations Per Second) – миллиард операций в секунду над числами с плавающей запятой (точкой).

Оценка производительности ЭВМ всегда приближительная, ибо при этом ориентируются на некоторые усредненные или, наоборот, на конкретные виды операций. Реально при решении различных задач используются и различные наборы операций. Поэтому для характеристики ПК вместо производительности обычно указывают тактовую частоту, более объективно определяющую быстродействие машины, так как каждая операция требует для своего выполнения вполне определенного количества тактов. Зная тактовую частоту, можно достаточно точно определить время выполнения любой машинной операции.

2. Разрядность машины и кодовых шин интерфейса.

Разрядность – это максимальное количество разрядов двоичного числа, над которым одновременно может выполняться машинная операция, в том числе и операция передачи информации; чем больше разрядность, тем, при прочих равных условиях, будет больше и производительность ПК.

3. Типы системного и локальных интерфейсов.

Разные типы интерфейсов обеспечивают разные скорости передачи информации между узлами машины, позволяют подключать разное количество внешних устройств и различные их виды.

4. Емкость оперативной памяти.

Емкость оперативной памяти измеряется чаще всего в мегабайтах (Мб), реже в килобайтах (кб). Напоминаем: $1 \text{ Мб} = 1024 \text{ кб} = 1048576 \text{ байт}$.

Многие современные прикладные программы при оперативной памяти емкостью меньше 8 Мб просто не работают либо работают, но очень медленно.

Следует иметь в виду, что увеличение емкости основной памяти в 2 раза, помимо всего прочего, дает повышение эффективной производительности ЭВМ при решении сложных задач примерно в 1,7 раза.

5. Емкость накопителя на жестких магнитных дисках (винчестер).

Емкость винчестера измеряется обычно в мегабайтах или гигабайтах ($1 \text{ Гб} = 1024 \text{ Мб}$).

6. Тип и емкость накопителей на гибких магнитных дисках.

Сейчас применяются в основном накопители на гибких магнитных дисках, использующие дискеты диаметром 3,5 и 5,25 дюйма (1 дюйм = 25,4 мм). Первые имеют стандартную емкость 1,44 Мб, вторые – 1,2 Мб.

7. Виды и емкость КЭШ-памяти.

КЭШ-память – это буферная, не доступная для пользователя быст-родействующая память, автоматически используемая компьютером для ускорения операций с информацией, хранящейся в более медленно действующих запоминающих устройствах. Например, для ускорения операций с основной памятью организуется регистровая КЭШ-память внутри микропроцессора (КЭШ-память первого уровня) или вне микропроцессора на материнской плате (КЭШ-память второго уровня); для ускорения операций с дисковой памятью организуется КЭШ-память на ячейках электронной памяти.

Следует иметь в виду, что наличие КЭШ-памяти емкостью 256 кб увеличивает производительность ПК примерно на 20 %.

8. Тип видеомонитора (дисплея) и видеоадаптера.

9. Тип принтера.

10. Наличие математического сопроцессора.

Математический сопроцессор позволяет в десятки раз ускорить выполнение операций над двоичными числами с плавающей запятой и над двоично-кодированными десятичными числами.

11. Имеющееся программное обеспечение и вид операционной системы.

12. Аппаратная и программная совместимость с другими типами ЭВМ.

Аппаратная и программная совместимость с другими типами ЭВМ означает возможность использования на компьютере, соответственно, тех же технических элементов и программного обеспечения, что и на других типах машин.

13. Возможность работы в вычислительной сети.

14. Возможность работы в многозадачном режиме.

Многозадачный режим позволяет выполнять вычисления одновременно по нескольким программам (многопрограммный режим) или для нескольких пользователей (многопользовательский режим). Совмещение во времени работы нескольких устройств машины, возможное в таком режиме, позволяет значительно увеличить эффективное быстродействие ЭВМ.

15. Надежность.

Надежность – это способность системы выполнять полностью и правильно все заданные ей функции. Надежность ПК измеряется обычно средним временем наработки на отказ.

16. Стоимость.

17. Габариты и масса.

3.6. Система команд микропроцессора

Решение задач на ЭВМ реализуется программным способом, т. е. путем выполнения последовательно во времени отдельных операций над информацией, предусмотренных алгоритмом решения задачи.

Алгоритм – это точно определенная последовательность действий, которые необходимо выполнить над исходной информацией, чтобы получить решение задачи.

Алгоритм решения задачи, заданный в виде последовательности команд на языке вычислительной машины (в кодах машины), называется *машинной программой* [2, 6].

Проектирование системы команд оказывает влияние на структуру ЭВМ. Оптимальную систему команд иногда определяют как совокупность команд, которая удовлетворяет требованиям проблемно-ориентированных применений таким образом, что избыточность аппаратных и аппаратно-программных средств на реализацию редко используемых команд оказывается минимальной. В различных программах ЭВМ частота появления команд различна; например, по данным фирмы DEC в программах для ЭВМ семейства PDP-11 наиболее часто встречается команда передачи MOV(B), на ее долю приходится приблизительно 32 % всех команд в типичных программах. Систему команд следует выбирать таким образом, чтобы затраты на редко используемые команды были минимальными.

При наличии статистических данных можно разработать (выбрать) ЭВМ с эффективной системой команд. Одним из подходов к достижению данной цели является разработка команд длиной в одно слово и кодирование их таким образом, чтобы разряды таких коротких команд использовать оптимально, что позволит сократить время реализации программы и ее длину.

Другим подходом к оптимизации системы команд является использование микроинструкций. В этом случае отдельные биты или группы бит команды используются для кодирования нескольких элементарных операций, которые выполняются в одном командном цикле.

Эти элементарные операции не требуют обращения к памяти, а последовательность их реализации определяется аппаратной логикой.

Сокращение времени выполнения программ и емкости памяти достигается за счет увеличения сложности логики управления.

Важной характеристикой команды является ее формат, определяющий структурные элементы команды, каждый из которых интерпретируется определенным образом при ее выполнении. Среди таких элементов (полей) команды выделяют следующие: *код операции*, определяющий выполняемое действие; *адрес* ячейки памяти, регистра процессора, внешнего устройства; режим адресации; операнд при использовании непосредственной адресации; код анализируемых признаков для команд условного перехода [2, 6].

Классификация команд по основным признакам представлена на рис. 6.

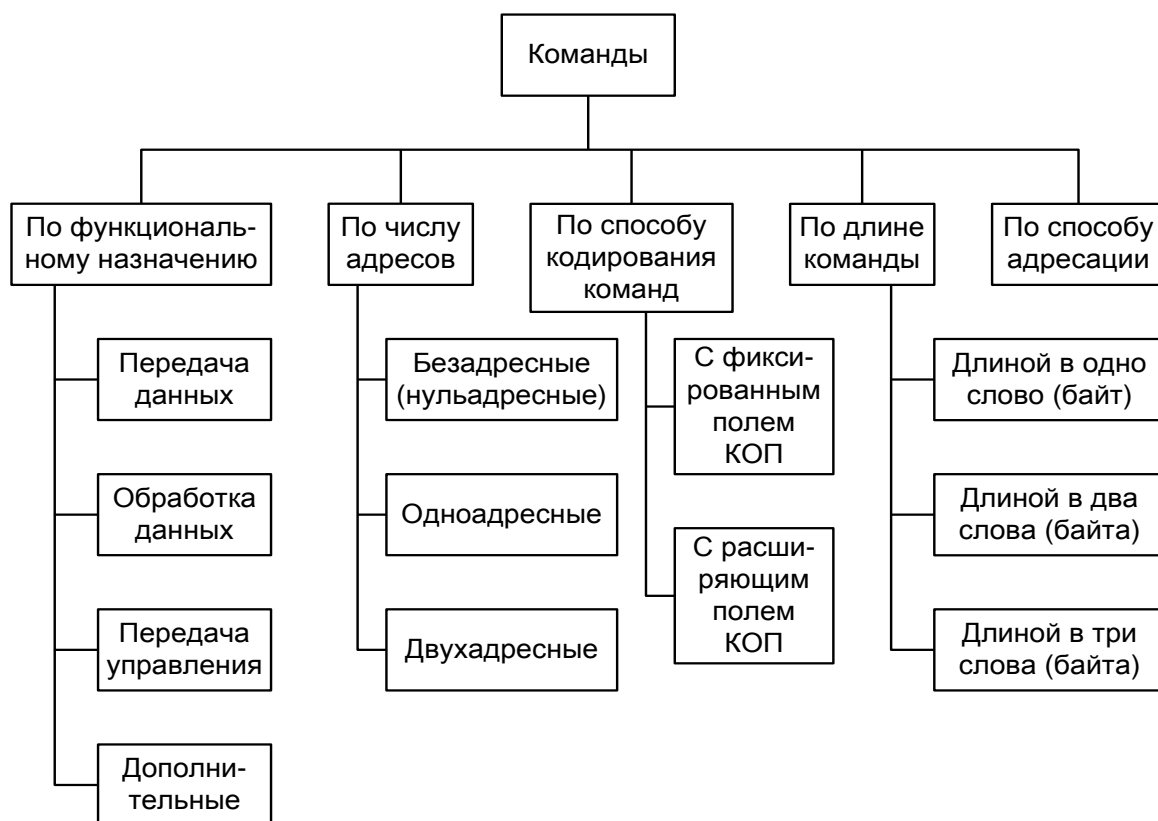


Рис. 6. Классификация команд

Важнейшим структурным элементом формата любой команды является *код операции* (КОП), определяющей действие, которое должно быть выполнено. Большое число КОП в процессоре очень важно, так как аппаратная реализация команд экономит память и время. Но при

выборе ЭВМ необходимо концентрировать внимание на полноте операций с конкретными типами данных, а не только на числе команд, на доступных режимах адресации. Число бит, отводимое под КОП, является функцией полного набора реализуемых команд [3, 6].

При использовании фиксированного числа бит под КОП для кодирования всех команд необходимо в поле КОП выделить m двоичных разрядов. Однако, учитывая ограниченную длину слова мини- и микроЭВМ, различное функциональное назначение команд, источники и приемники результатов операций, а также то, что не все команды содержат адресную часть для обращения к памяти и периферийным устройствам, в малых ЭВМ для кодирования команд широко используется принцип кодирования с переменным числом бит под поле КОП для различных групп команд.

В некоторых командах необходим только один операнд и они называются *однооперандными* (или одноадресными) командами в отличие от *двухоперандных* (или двухадресных), в которых требуются два операнда. При наличии двух операндов командой обычно изменяется только один из них. Так как информация берется только из одной ячейки, эту ячейку называются *источником*; ячейка, содержимое которой изменяется, называется *приемником*.

3.6.1. Структура и виды команд

Команда машинной программы (иначе, машинная команда) – это элементарная инструкция машине, выполняемая ею автоматически без каких-либо дополнительных указаний и пояснений.

Машинная команда состоит из двух частей: операционной и адресной.

Операционная часть команды – это группа разрядов в команде, предназначенная для представления кода операции машины.

Адресная часть команды – это группа разрядов в команде, в которой записываются коды адреса (адресов) ячеек памяти машины, предназначенных для оперативного хранения информации или иных объектов, задействованных при выполнении команды. Часто эти адреса называются адресами операндов, т. е. чисел, участвующих в операции.

По количеству адресов, записываемых в команде, команды делятся на безадресные, одно-, двух- и трехадресные.

Типовая структура трехадресной команды [2, 3, 6]:

$$\text{КОП } a_1, a_2, a_3,$$

где a_1 и a_2 – адреса ячеек (регистров), где расположены, соответственно, первое и второе числа, участвующие в операции;

a3 – адрес ячейки (регистра), куда следует поместить число, полученное в результате выполнения операции.

Типовая структура двухадресной команды:

КОП a1, a2,

где a1 – это обычно адрес ячейки (регистра), где хранится первое из чисел, участвующих в операции, и куда после завершения операции должен быть записан результат операции;

a2 – обычно адрес ячейки (регистра), где хранится второе участвующее в операции число.

Типовая структура одноадресной команды:

КОП a1,

где a1 в зависимости от модификации команды может обозначать либо адрес ячейки (регистра), где хранится одно из чисел, участвующих в операции, либо адрес ячейки (регистра), куда следует поместить число результат операции.

Безадресная команда содержит только код операции, а информация для нее должна быть заранее помещена в определенные регистры машины (безадресные команды могут использоваться только совместно с командами другой адресности).

Пример. Поступила представленная на языке символического кодирования команда:

СЛ 0103, 5102.

Такую команду следует расшифровать так: "сложить число, записанное в ячейке 0103 памяти, с числом, записанным в ячейке 5102, а затем результат (т. е. сумму) поместить в ячейку 0103".

3.6.2. Особенности программного и микропрограммного управления

В микропроцессорах используют два метода выработки совокупности функциональных управляющих сигналов: программный и микропрограммный [2, 3, 6].

Выполнение операций в машине сводится к элементарным преобразованиям информации (передача информации между узлами в блоках, сдвиг информации в узлах, логические поразрядные операции, проверка условий и т. д.) в логических элементах, узлах и блоках под воздействием функциональных управляющих сигналов блоков (устройств) управления. Элементарные преобразования, неразложимые на более простые, выполняются в течение одного такта сигналов синхронизации и называются микрооперациями.

В аппаратных (схемных) устройствах управления каждой операции соответствует свой набор логических схем, вырабатывающих определенные функциональные сигналы для выполнения микроопераций в определенные моменты времени. При этом способе построения устройства управления реализация микроопераций достигается за счет однажды соединенных между собой логических схем, поэтому ЭВМ с аппаратным устройством управления называют ЭВМ с жесткой логикой управления. Это понятие относится к фиксации системы команд в структуре связей ЭВМ и означает практическую невозможность каких-либо изменений в системе команд ЭВМ после ее изготовления.

При микропрограммной реализации устройства управления в состав последнего вводится ЗУ, каждый разряд выходного кода которого определяет появление определенного функционального сигнала управления. Поэтому каждой микрооперации ставится в соответствие свой информационный код – микрокоманда. Набор микрокоманд и последовательность их реализации обеспечивают выполнение любой сложной операции. Набор микроопераций называют микропрограммами. Способ управления операциями путем последовательного считывания и интерпретации микрокоманд из ЗУ (наиболее часто в виде микропрограммного ЗУ используют быстродействующие программируемые логические матрицы), а также использования кодов микрокоманд для генерации функциональных управляющих сигналов называют микропрограммным, а микроЭВМ с таким способом управления – микропрограммными или с хранимой (гибкой) логикой управления.

К микропрограммам предъявляют требования функциональной полноты и минимальности. Первое требование необходимо для обеспечения возможности разработки микропрограмм любых машинных операций, а второе связано с желанием уменьшить объем используемого оборудования. Учет фактора быстродействия ведет к расширению микропрограмм, поскольку усложнение последних позволяет сократить время выполнения команд программы.

Преобразование информации выполняется в универсальном арифметико-логическом блоке микропроцессора. Он обычно строится на основе комбинационных логических схем.

Для ускорения выполнения определенных операций вводятся дополнительно специальные операционные узлы (например, узел циклического сдвига). Кроме того, в состав микропроцессорного комплекта (МПК) БИС вводятся специализированные оперативные блоки арифметических расширителей.

Операционные возможности микропроцессора можно расширить за счет увеличения числа регистров. Если в регистровом буфере закре-

пение функций регистров отсутствует, то их можно использовать как для хранения данных, так и для хранения адресов. Подобные регистры микропроцессора называются регистрами общего назначения (РОН). По мере развития технологии реально осуществлено изготовление в микропроцессоре 16, 32 и более регистров.

В целом же, принцип микропрограммного управления (ПМУ) включает следующие позиции:

1) любая операция, реализуемая устройством, является последовательностью элементарных действий – микроопераций;

2) для управления порядком следования микроопераций используются логические условия;

3) процесс выполнения операций в устройстве описывается в форме алгоритма, представляемого в терминах микроопераций и логических условий, называемого микропрограммой;

4) микропрограмма используется как форма представления функции устройства, на основе которой определяются структура и порядок функционирования устройства во времени.

ПМУ обеспечивает гибкость микропроцессорной системы и позволяет осуществлять проблемную ориентацию микро- и миниЭВМ.

3.7. Структура микропроцессора

3.7.1. Типовая структура микропроцессора

Типовая структура микропроцессора [1, 3, 7] приведена на рис. 7. Микропроцессор состоит из трех основных блоков: арифметико-логическое устройство (АЛУ), блок внутренних регистров (микропроцессорная память) и устройство управления. Для передачи данных между этими блоками используется внутренняя шина данных.

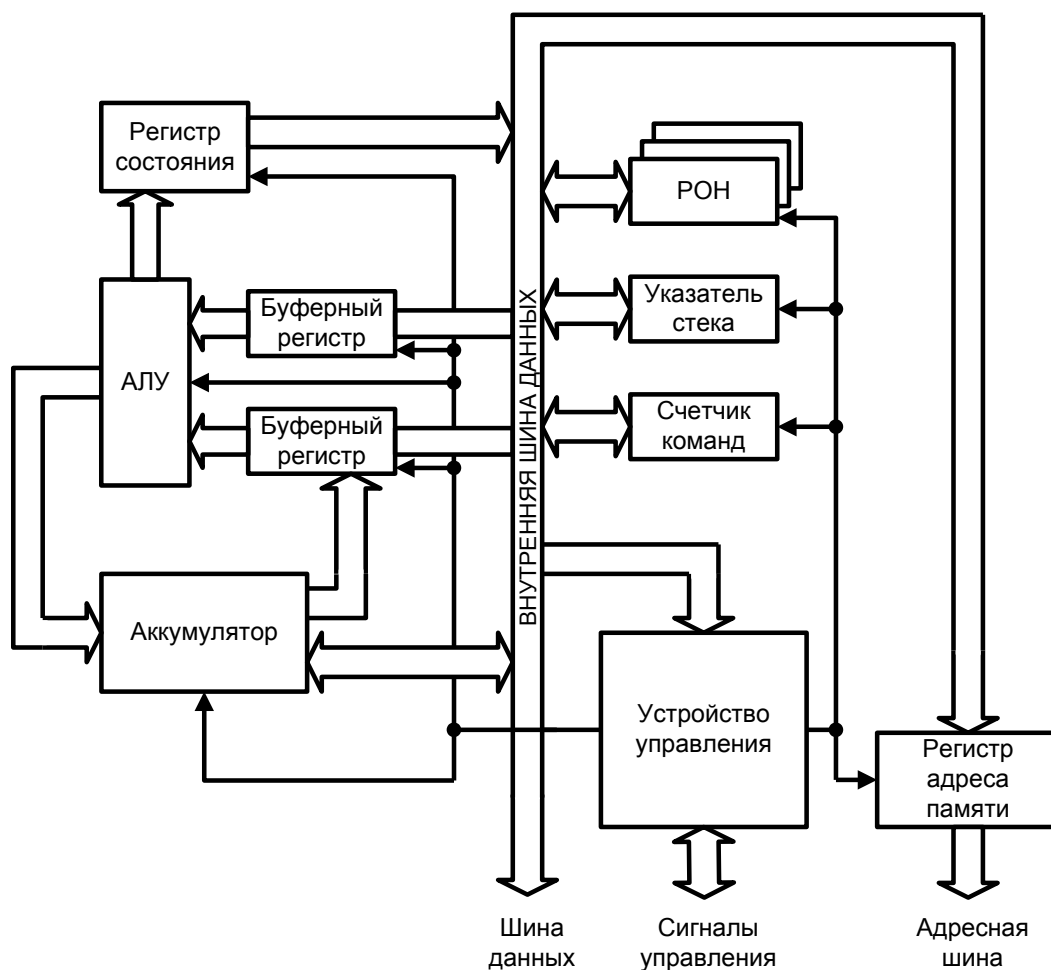


Рис. 7. Типовая структурная схема микропроцессора

Арифметико-логическое устройство выполняет одну из главных функций микропроцессора – обработку данных [7]. Перечень функций АЛУ зависит от типа микропроцессора. Некоторые АЛУ способны вы-

полнять множество различных операций, у других набор операций ограничен. Функции АЛУ определяют архитектуру микропроцессора в целом. Операции, выполняемые АЛУ, большинства микропроцессоров следующие: сложение, вычитание, И, ИЛИ, исключающее ИЛИ, инверсия, сдвиг вправо, сдвиг влево, приращения положительное и отрицательное.

Важная составная часть микропроцессора – регистры. Каждый регистр микропроцессора можно использовать для временного хранения одного слова данных. Некоторые регистры имеют *специальное назначение*, другие – многоцелевое. Последние называются регистрами *общего назначения* (РОН) и могут использоваться программистом по его усмотрению. Количество и назначение регистров в микропроцессоре зависят от его архитектуры.

Рассмотрим назначение основных регистров, имеющих почти во всех микропроцессорах.

Аккумулятор – это главный регистр микропроцессора при различных манипуляциях с данными [7]. Большинство арифметических и логических операций осуществляется путем использования АЛУ и аккумулятора. Любая из таких операций над двумя словами данных (операндами) предполагает размещение одного из них в аккумуляторе, а другого в памяти или каком-либо регистре. Так, при сложении двух слов, называемых условно А и В и расположенных в аккумуляторе и памяти соответственно, результирующая сумма С загружается в аккумулятор, замещая слово А. Результат выполнения операции АЛУ тоже обычно размещается в аккумуляторе, содержимое которого при этом теряется.

Операцией другого типа, использующей аккумулятор, является программируемая передача данных из одной части микропроцессора в другую. Например, пересылка данных между портом ввода/вывода и памятью, между двумя областями памяти и т. д. Выполнение операции «программируемая передача данных» осуществляется в два этапа: сначала выполняется пересылка данных из источника в аккумулятор, затем – из аккумулятора в пункт назначения.

Микропроцессор может выполнять некоторые действия над данными непосредственно в аккумуляторе. Например, аккумулятор можно очистить путем записи двоичных нулей во все его разряды, установить в единичное состояние путем записи во все его разряды двоичных единиц. Содержимое аккумулятора можно сдвигать влево или вправо, получать его инвертированное значение, а также выполнять другие операции. Аккумулятор является наиболее универсальным регистром микропроцессора: для выполнения любой операции над данными пре-

жде всего необходимо поместить их в аккумулятор. Данные поступают в него с внутренней шины данных микропроцессора. В свою очередь, аккумулятор может посылать данные на эту шину.

Количество разрядов аккумулятора соответствует длине слова микропроцессора, однако некоторые микропроцессоры имеют аккумуляторы двойной длины. В дополнительные разряды аккумулятора записываются при этом биты, появляющиеся при выполнении некоторых арифметических операций. Например, при умножении двух 8-битовых слов результат (16-битовое число) размещается в аккумуляторе двойной длины.

Счетчик команд – это один из наиболее важных регистров микропроцессора [7]. Как известно, *программа* – это последовательность команд (инструкций), хранимых в памяти микроЭВМ и предназначенных для того, чтобы инструктировать машину, как решать поставленную задачу. Для корректного ее выполнения команды должны поступать в строго определенном порядке. Счетчик команд обеспечивает формирование адреса очередной команды, записанной в памяти.

Когда микропроцессор начинает работать, то по команде начальной установки в счетчик команд загружаются данные из области памяти, заданной проектировщиком микропроцессора. Когда программа начинает выполняться, первым значением содержимого счетчика команд является этот, заранее определенный адрес.

В отличие от аккумулятора счетчик команд не может выполнять операции различного типа. Набор команд, его использующих, крайне ограничен по сравнению с подобным набором для аккумулятора.

Перед выполнением программы счетчик команд необходимо загрузить адресом, указывающим на первую команду программы. Адрес первой команды программы посылается по адресной шине к схемам управления памятью, в результате чего считывается ее содержимое по указанному адресу. Далее эта команда передается в специальный регистр микропроцессора, называемый *регистром команд* [7].

После извлечения команды из памяти микропроцессор автоматически дает приращение содержимому счетчика команд. Это приращение счетчик команд получает в тот момент, когда микропроцессор начинает выполнять команду, только что извлеченную из памяти. Следовательно, с этого момента счетчик команд содержит адрес следующей команды.

Счетчик команд можно загрузить иным содержимым при выполнении особой группы команд. Может возникнуть необходимость выполнить часть программы, которая «выпадает» из последовательности команд основной (главной) программы. Например, такую часть про-

граммы, которая повторяется в процессе выполнения всей программы. Вместо того чтобы писать эту часть программы каждый раз, когда в ней возникает необходимость, ее записывают один раз и возвращаются к ее повторному выполнению, отступая от указанной последовательности. Часть программы, выполняемая путем отступления от последовательности команд главной программы, называется подпрограммой. В данном случае в счетчик команд непосредственно записывается требуемый адрес.

Часто счетчик команд имеет намного больше разрядов, чем длина слова данных микропроцессора. Так, в большинстве 8-разрядных микропроцессоров число разрядов счетчика команд равно 16.

Регистр команд содержит команду в процессе ее дешифрования и выполнения. Входные данные поступают в регистр из памяти по мере последовательной выборки команд. Обычно существует возможность записи данных в регистр команд при помощи набора переключателей и кнопок на пульте управления ЭВМ. Как правило, этой возможностью пользуются для передачи управления в начало программы.

Регистр адреса памяти при каждом обращении к памяти микроЭВМ указывает адрес области памяти, подлежащей использованию микропроцессором. Регистр адреса памяти содержит двоичное число – адрес области памяти. Выход этого регистра называется **адресной шиной** и используется для выбора области памяти или порта ввода/вывода [7].

В течение выборки команды из памяти регистры адреса памяти и счетчика команд имеют одинаковое содержимое, т. е. регистр адреса памяти указывает местоположение команды, извлекаемой из памяти.

После декодирования команды счетчик команд получает приращение в отличие от регистра адреса памяти.

В процессе выполнения команды содержимое регистра адреса памяти зависит от выполняемой команды. Если в соответствии с командой микропроцессор должен произвести еще одно обращение к памяти, то регистр адреса памяти подлежит вторичному использованию в процессе обработки этой команды. Для некоторых команд, например команды очистки аккумулятора, адресация к памяти не требуется. При обработке таких команд регистр адреса памяти используется лишь один раз – в течение выборки команды из памяти.

В большинстве микропроцессоров регистры адреса памяти и счетчика команд имеют одинаковое количество разрядов. Как и счетчик команд, регистр адреса памяти должен располагать количеством разрядов, достаточным для адресации любой области памяти микроЭВМ. У

большинства 8-разрядных микропроцессоров количество разрядов регистра адреса памяти равно 16.

Поскольку регистр адреса памяти подключен к внутренней шине данных микропроцессора, он может загружаться от различных источников. Большинство микропроцессоров располагают командами, позволяющими загружать этот регистр содержимым счетчика команд, регистра общего назначения или какой-либо области памяти. Некоторые команды предоставляют возможность изменять содержимое регистра адреса памяти путем выполнения вычислений: новое значение содержимого этого регистра получается путем сложения или вычитания содержимого счетчика команд с числом, указанным в самой команде. Адресация такого типа называется адресацией с использованием смещения.

Буферный регистр предназначен для временного хранения (буферирования) данных [7].

Регистр состояния предназначен для хранения результатов некоторых проверок, осуществляемых в процессе выполнения программы. Разряды регистра состояний принимают то или иное значение при выполнении операций, использующих АЛУ и некоторые регистры. Запоминание результатов упомянутых проверок позволяет использовать программы, содержащие переходы (нарушения естественной последовательности выполнения команд) [7].

При наличии в программе перехода по заданному признаку выполнение команд начинается с некоторой новой области памяти, т. е. счетчик команд загружается новым числом. В случае условного перехода такое действие имеет место, если результаты определенных проверок совпадают с ожидаемыми значениями. Указанные результаты находятся в регистре состояния. Регистр состояния предоставляет программисту возможность организовать работу микропроцессора так, чтобы при определенных условиях менялся порядок выполнения команд.

Рассмотрим некоторые наиболее часто используемые разряды регистра состояния.

1. Перенос/заем. Данный разряд указывает, что последняя выполненная операция сопровождалась переносом или займом (отрицательным переносом). Значение разряда переноса устанавливается равным 1, если в результате сложения двух чисел имеет место перенос из старшего разряда АЛУ. Отрицательный перенос (заем) фиксируется в регистре состояния при вычитании большего числа из меньшего.

2. Нулевой результат. Принимает единичное значение, если после окончания операции во всех разрядах регистра результата обнаружены

двоичные нули. Установка этого разряда в 1 происходит не только при отрицательном приращении содержимого регистра, но и при любой другой операции, результатом которой – число из двоичных нулей.

3. Знаковый. Принимает единичное значение, когда старший значащий бит содержимого регистра, предназначенного для записи результата операции, становится равным 1. При выполнении арифметических операций с числами в дополнительном коде единичное значение старшего значащего бита показывает, что в регистре находится отрицательное число. Многие микропроцессоры располагают дополнительными разрядами состояний. В некоторых предусмотрены специальные команды для сброса или очистки всех разрядов состояния.

Регистры общего назначения (РОН) [7]. Большинство МП имеют в своем составе набор регистров, используемых в качестве запоминающих устройств. Так как АЛУ может совершать операции с содержимым РОН без выхода на внешнюю магистраль адресов и данных, то они происходят много быстрее, чем операции с внешней памятью. Поэтому иногда РОН называют сверхоперативной памятью. Количество РОН и возможность программного доступа к ним у равных микропроцессоров различны.

Указатель стека. *Стек* – это набор регистров микропроцессора или ячеек оперативной памяти, откуда данные или адреса выбираются «сверху» по принципу: первым – поступивший последним. При записи в стек очередного слова все ранее записанные слова смещаются на один регистр вниз. При выборке слова из стека оставшиеся слова перемещаются на один регистр вверх.

Указанные процедуры иллюстрирует рис. 8 [7]. Здесь стек состоит из семи регистров. Если в стек загружается какое-либо слово, например А5, то оно записывается в верхнем регистре, а каждое из слов А1 ... А4 перемещается в соседние нижние регистры. Если же А5 извлекается из стека, то каждое из слов А1 ... А4 перемещается в соседние верхние регистры. Нельзя извлечь А4 раньше А5, т. е. автоматически реализуется отмеченный выше принцип. Стек обычно используется в микропроцессорах для хранения адресов возврата при обращении к подпрограммам, а также для запоминания состояния внутренних регистров при обработке прерываний. При организации стека в памяти время на обращение к нему будет равно циклу обращения к памяти. Эта операция выполняется значительно быстрее, если стек в виде набора регистров входит в состав микропроцессора. Важным параметром в таком случае является число регистров стека. При попытке записать в стек большее количество слов, чем число его регистров, первое слово будет утеряно. В некоторых микропроцессорах при переполнении ре-

гистров стека соответствующие слова записываются в стек памяти. Таким образом, процесс его функционирования напоминает работу с пачкой документов, когда каждый новый документ кладется сверху пачки.

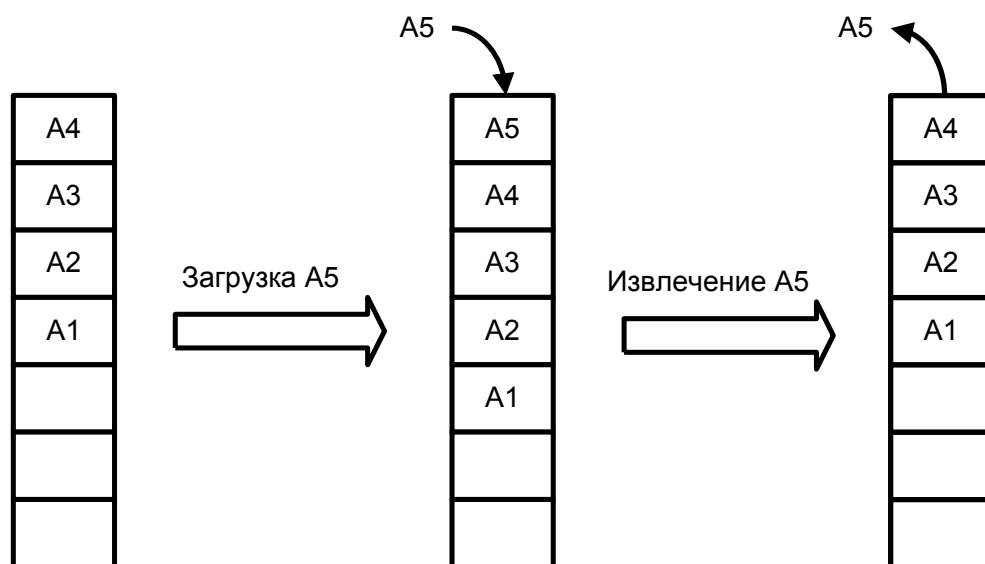


Рис. 8. Принципы работы стека

В приведенном примере при записи в стек новых данных все его содержимое перемещается на один регистр. В реальных МП такой способ не применяется, так как он сопряжен с большим количеством вспомогательных операций по переносу соседних регистров стека.

В реальных МП данные между ячейками не переносятся, а для хранения адреса последнего по времени поступления элемента стека используется специальный регистр – указатель стека (УС).

На рис. 9 указатель стека представляет собой трехразрядный регистр с двоичным представлением информации [7].

Первоначально указатель стека содержит число 0112. Это означает, что последний элемент – «верхушка стека» – находится в регистре с адресом 0112 (или 310). При операции загрузки в регистр 410 записывается число A5, а содержимое указателя стека изменяется так, что он указывает на регистр 410. При операции извлечения из стека производятся обратные действия.

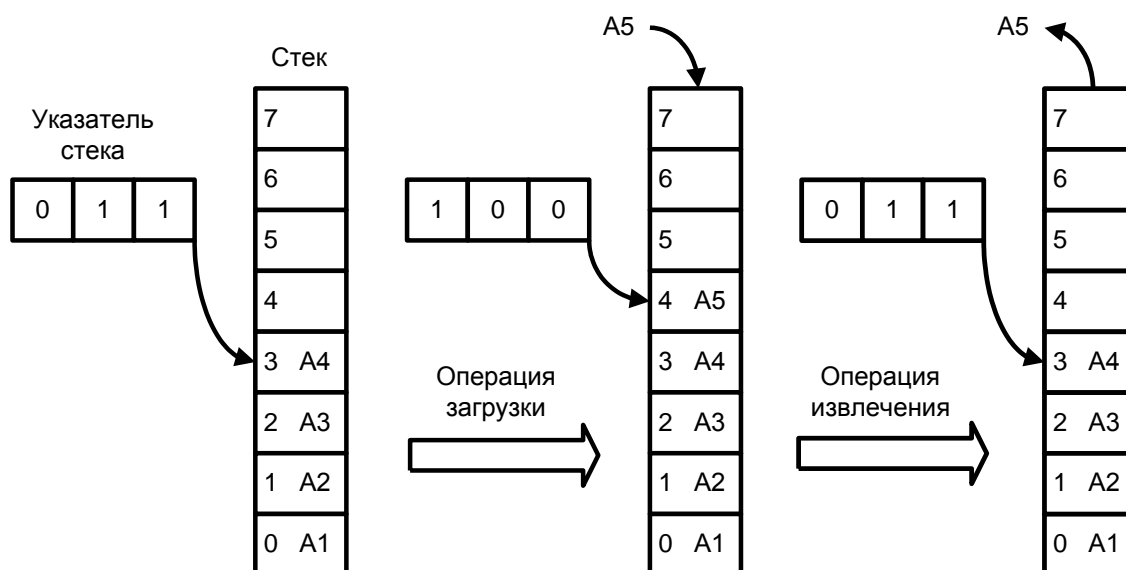


Рис. 9. Адресация элемента стека с использованием указателя стека

Схемы управления. Роль схем управления в микропроцессоре заключается в поддержании требуемой последовательности функционирования всех остальных его звеньев. По сигналам схем управления очередная команда извлекается из регистра команд. При этом определяется, что необходимо делать с данными, а затем обеспечивается последовательность действий для выполнения поставленной задачи.

Одна из главных функций схем управления – декодирование команды, находящейся в регистре команд, посредством дешифратора команд, который в результате выдает сигналы, необходимые для ее выполнения.

Помимо указанных выше действий, схемы управления выполняют некоторые специальные функции: управление последовательностью включения питания и процессами прерываний. *Прерывание* – это своего рода запрос, поступающий на схемы управления от других устройств (памяти, ввода/вывода). Прерывание связано с использованием внутренней шины данных микропроцессора. Схемы управления принимают решение, когда и в какой последовательности другие устройства могут пользоваться внутренней шиной данных.

Система шин. На характеристики микропроцессора большое влияние оказывает способ организации его связи с внешней средой – устройствами ввода/вывода (УВВ) и запоминающими устройствами (ЗУ). По способу организации связей с внешней средой различают микропроцессоры с мультиплексированной шиной адреса и данных (рис. 10,а) и с отдельными шинами адреса и данных (рис. 10,б). Мик-

ропроцессор с отдельными шинами адресов и данных изображен на рис. 7.

В микропроцессорах с мультиплексированной шиной адрес хранится на шине только короткий промежуток времени, поэтому устройствам, подключенным к шине, требуются регистры адреса (РгА). Для организации обмена информацией в таких микропроцессорах необходимо использовать управляющий сигнал *адрес–данные*. При отдельных шинах адреса и данных такой управляющий сигнал не нужен. Кроме того, у устройств, подключенных к шинам, отпадает необходимость в регистре адреса, так как он может быть размещен непосредственно на кристалле микропроцессора. Разрядность адресной шины в таких микропроцессорах не связана с разрядностью шины данных.

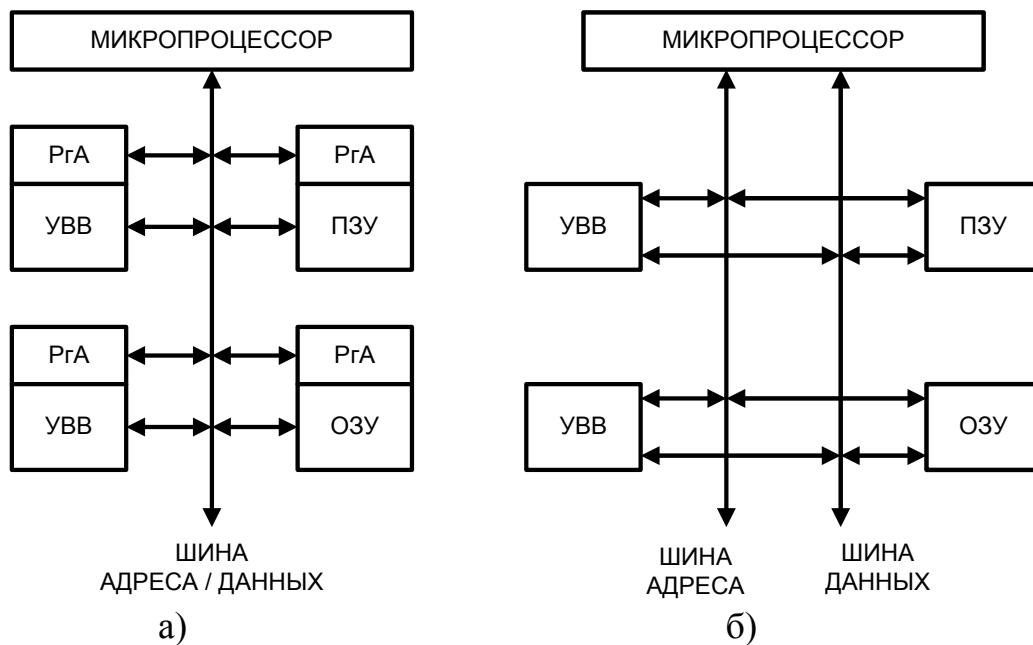


Рис. 10. Система шин микроЭВМ

3.7.2. Устройство управления

Основные функции:

1. Формирование адреса инструкции.
2. Считывание инструкции из ОЗУ (ПЗУ) и её хранение во время выполнения.
3. Дешифрация кода операции.
4. Формирование управляющих сигналов.
5. Считывание из регистра команд и регистров микропроцессорной памяти отдельных составляющих адресов операндов (чисел), участвующих в вычислениях, и формирование полных адресов операндов.
6. Выборка операндов (по сформированным адресам) и выполнение заданной операции обработки этих операндов.
7. Запись результатов операции в память.
8. Формирования адреса следующей команды программы.

Упрощенная структурная схема устройства управления (УУ) показана на рис. 11 [3, 4, 7].

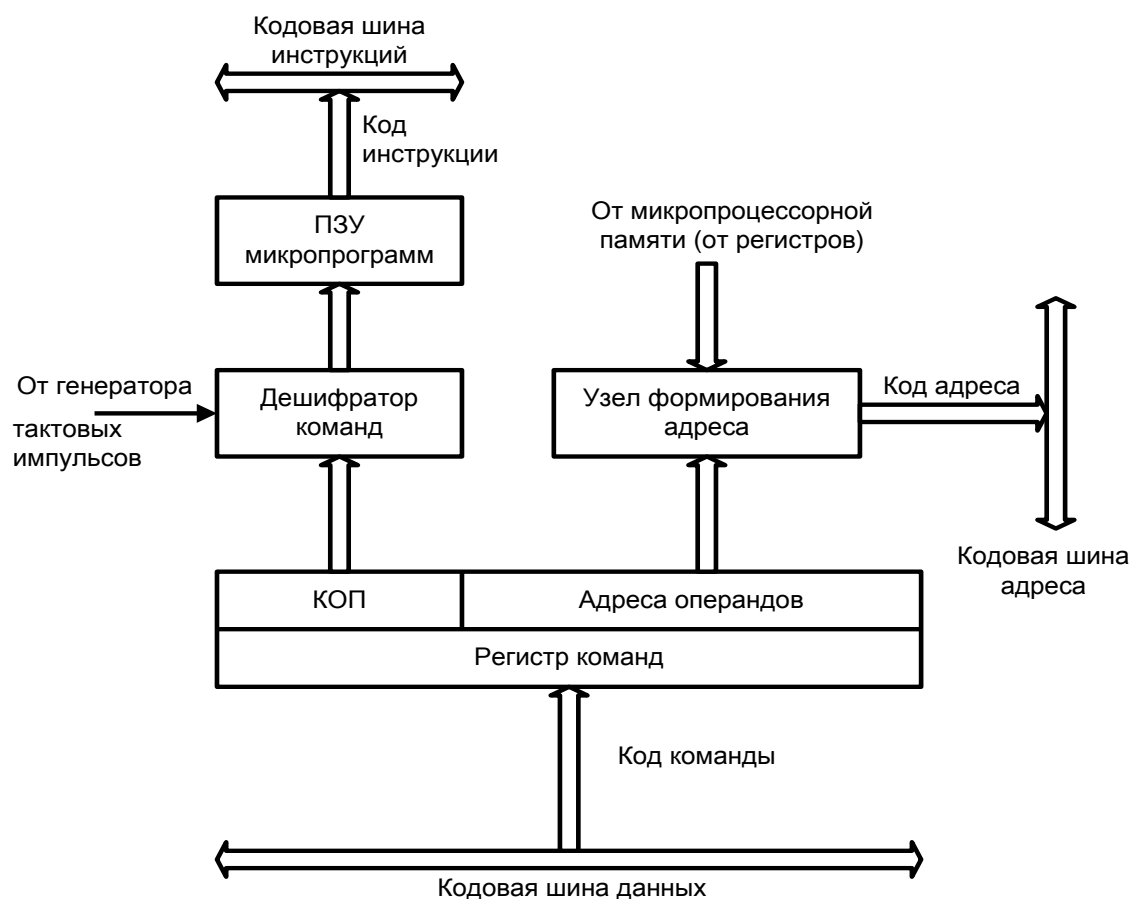


Рис. 11. Упрощённая структурная схема устройства управления

Состав структурной схемы:

Регистр команд – запоминающий регистр, в котором хранится код команды: код выполняемой операции и адреса операндов, участвующих в операции.

Дешифратор операций – логический блок, выбирающий из регистра команд код операции (КОП).

Постоянное запоминающее устройство микропрограмм – хранит в своих ячейках управляющие сигналы (импульсы), необходимые для выполнения в блоках ПК операций обработки информации. Импульс по выбранной дешифратором операций, в соответствии с кодом, операции считывает из ПЗУ микропрограмм необходимую последовательность управляющих сигналов.

Узел формирования адреса – устройство, вычисляющее полный адрес ячейки памяти (регистра) по реквизитам, поступающим из регистра команд и регистров микропроцессорной памяти.

Кодовые шины данных, адреса и инструкций – часть внутренней интерфейсной шины микропроцессора.

3.7.3. Арифметико-логическое устройство

Основные функции АЛУ:

1. Приём операндов из ОЗУ и регистров (микропроцессорной памяти).
2. Выполнение арифметических и логических операций.
3. Передача результатов в регистры и в ОЗУ.
4. Формирование признаков результатов выполнения операций и их запись в регистр слова состояния процессора.

В состав АЛУ входят сумматоры, которые непосредственно выполняют элементарные операции, регистры для хранения операндов и результатов, сдвиговые регистры, логические схемы И, ИЛИ, НЕ, схемы для преобразования прямого кода числа в дополнительную форму и дополнительные схемы аппаратного умножения и деления для целочисленных операндов.

Упрощенная структурная схема АЛУ показана на рис. 12 [3, 4, 7].

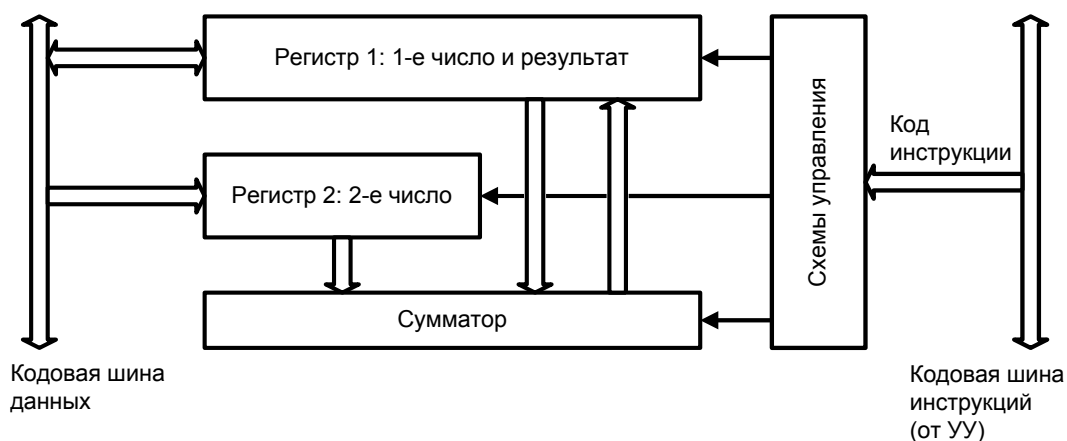


Рис. 12. Упрощённая структурная схема АЛУ

Состав структурной схемы:

Сумматор – вычислительная схема, выполняющая процедуру сложения поступающих на ее вход двоичных кодов; сумматор имеет разрядность двойного машинного слова.

Регистры – быстродействующие ячейки памяти различной длины: регистр 1 (Rг1) имеет разрядность двойного слова, а регистр 2 (Rг2) – разрядность слова.

При выполнении операций в Rг1 помещается первое число, участвующее в операции, а по завершении операции – результат; в Rг2 – второе число, участвующее в операции (по завершении операции информация в нем не изменяется). Регистр 1 может и принимать информацию с кодовых шин данных, и выдавать информацию на них, регистр 2 только получает информацию с этих шин.

Схемы управления принимают по кодовым шинам инструкций управляющие сигналы от устройства управления и преобразуют их в сигналы для управления работой регистров и сумматора АЛУ.

АЛУ выполняет арифметические операции (+, -, *, :) только над двоичной информацией с запятой, фиксированной после последнего разряда, т. е. только над целыми двоичными числами.

Выполнение операций над двоичными числами с плавающей запятой и над двоично-кодированными десятичными числами осуществляется или с привлечением математического сопроцессора, или по специально составленным программам.

Основные элементы АЛУ. Основой АЛУ является одnorазрядный сумматор, схема которого обеспечивает суммирование цифры одного разряда 2-го числа с учётом бита переноса из соседнего младшего раз-

ряда 1-го числа. Схема одноразрядного сумматора представлена на рис. 13.

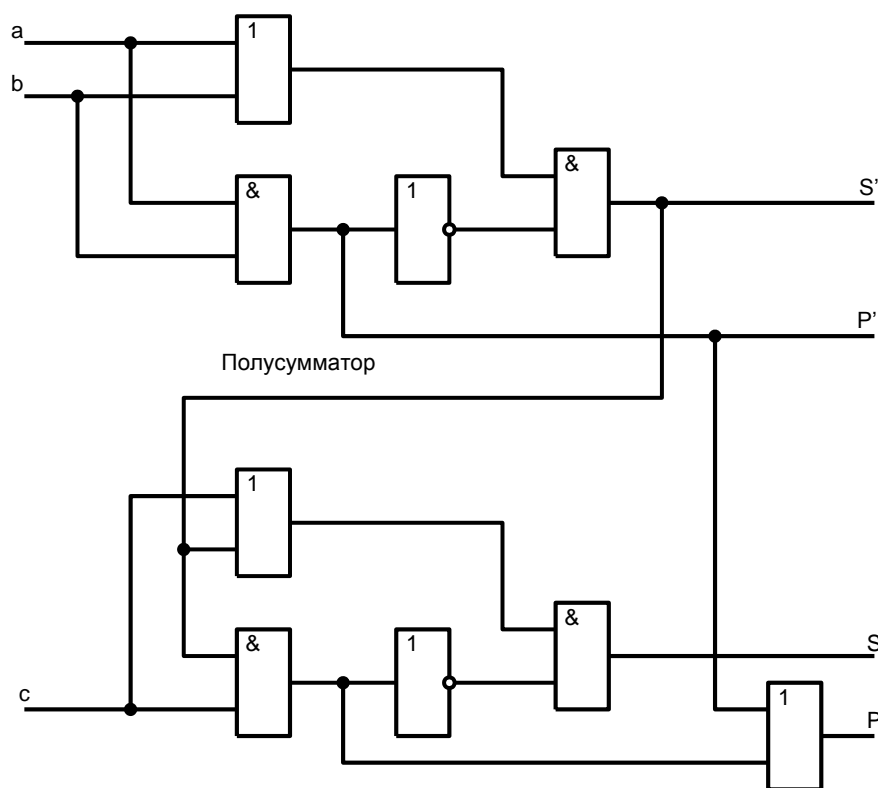


Рис. 13. Одноразрядный сумматор

Рассмотренный сумматор выполняет операцию сложения в соответствии с табл. 3.

Таблица 3

a	b	c	S'	P'	S	P
0	0	0	0	0	0	0
1	0	0	1	0	1	0
0	1	0	1	0	1	0
1	1	0	0	1	0	1
0	0	1	0	0	1	0
1	0	1	1	0	0	1
0	1	1	1	0	0	1
1	1	1	0	1	1	1

Примечание: а – соответствующий разряд 1-го числа;

- b – соответствующий разряд 2-го числа;
- c – бит переноса из соседнего младшего разряда;
- S' – значение цифры суммы в данном разряде полусумматора;
- P' – цифра переноса в следующий (старший) разряд полусумматора;
- S – значение цифры суммы в данном разряде;
- P – цифра переноса в следующий (старший) разряд.

Практически все операции в АЛУ сводятся к сложению и дополнительным операциям сдвига и преобразования кода.

3.7.4. Последовательность работы микропроцессора

После запуска в регистре счетчика команд микропроцессора устанавливается адрес ячейки ОЗУ, в которой находится начало (первая команда) программы.

После этого автоматически начинается выполнение команд программы друг за другом. Каждая команда требует для своего исполнения нескольких тактов работы машины (такты определяются периодом следования импульсов от генератора тактовых импульсов).

В первом такте выполнения любой команды производятся считывание кода самой команды из ОЗУ по адресу, установленному в регистре-счетчике команд, и запись этого кода в блок регистров команд устройства управления. Содержание второго и последующих тактов исполнения определяется результатами анализа команды, записанной в блок регистров команд, т. е. зависит уже от конкретной команды.

Пример. Выполнение команды сложения аккумулятора с прямоадресуемой ячейкой памяти ОЗУ (длина команды два байта) [1, 4, 8]:

add a, addr.

При выполнении данной команды будут выполнены следующие действия:

- Первый такт: считывание КОП из ОЗУ по адресу, установленному в регистре-счетчике команд, запись этого кода в блок регистров команд устройства управления, декодирование КОП, приращение регистра счётчика команд.
- Второй такт: запись содержимого аккумулятора в буферный регистр АЛУ.
- Третий такт: считывание из ячейки ОЗУ по адресу addr второго слагаемого и перемещение его во второй буферный регистр АЛУ, приращение регистра счётчика команд;

- Четвёртый такт: сложение в АЛУ переданных туда чисел и формирование суммы.
- Пятый такт: считывание из АЛУ суммы чисел и запись её в аккумулятор.

Далее МП приступит к выполнению следующей команды по адресу, содержащемуся в регистре-счётчике команд. Команды будут выполняться последовательно одни за другой, пока не завершится вся программа.

При наличии в программе перехода по заданному признаку выполнение команд начинается с некоторой новой области памяти, т. е. счётчик команд загружается новым числом. В случае условного перехода такое действие имеет место, если результаты определенных проверок совпадают с ожидаемыми значениями. Указанные результаты находятся в регистре состояния. Регистр состояния предоставляет программисту возможность организовать работу микропроцессора так, чтобы при определенных условиях менялся порядок выполнения команд.

3.7.5. Режимы адресации

Для взаимодействия с различными модулями в ЭВМ должны быть средства идентификации ячеек внешней памяти, ячеек внутренней памяти, регистров МП и регистров устройств ввода/вывода. Поэтому каждой из запоминающих ячеек присваивается адрес, т. е. однозначная комбинация бит. Количество бит определяет число идентифицируемых ячеек. Обычно ЭВМ имеет различные адресные пространства памяти и регистров МП, а иногда – отдельные адресные пространства регистров устройств ввода/вывода и внутренней памяти. Кроме того, память хранит как данные, так и команды. Поэтому для ЭВМ разработано множество способов обращения к памяти, называемых режимами адресации.

Режим адресации памяти – это процедура или схема преобразования адресной информации об операнде в его исполнительный адрес.

Все способы адресации памяти можно разделить на:

1) прямой, когда исполнительный адрес берется непосредственно из команды или вычисляется с использованием значения, указанного в команде, и содержимого какого-либо регистра (прямая адресация, регистровая, базовая, индексная и т. д.);

2) косвенный, который предполагает, что в команде содержится значение косвенного адреса, т. е. адреса ячейки памяти, в которой находится окончательный исполнительный адрес (косвенная адресация).

В каждой микроЭВМ реализованы только некоторые режимы адресации, использование которых, как правило, определяется архитектурой МП.

4. Память микропроцессорной системы

Для хранения информации в микропроцессорных системах используются запоминающие устройства на основе полупроводниковых материалов, а также магнитные и оптические внешние носители. Внутренняя память компьютера представлена в виде отдельных интегральных микросхем (ИМС), собственно памяти и элементов, включенных в состав других ИМС, не выполняющих непосредственно функцию хранения программ и данных – это и внутренняя память центрального процессора, и видеопамять, и контроллеры различных устройств [2, 5, 6, 9].

Для создания элементов запоминающих устройств, в основном, применяют СБИС со структурой МДП (металл–диэлектрик–полупроводник) на основе кремния (в связи с тем, что в качестве диэлектрика чаще всего используют его оксид SiO_2 , то их обычно называют МОП-структурами (металл–оксид–полупроводник)).

Для функционирования компьютерной системы необходимо наличие как оперативного запоминающего устройства (ОЗУ), так и постоянного запоминающего устройства (ПЗУ), обеспечивающего сохранение информации при выключении питания. ОЗУ может быть статическим и динамическим, а ПЗУ – однократно или многократно программируемым.

Степень интеграции, быстродействие, электрические параметры ЗУ при записи и хранении информации, помехоустойчивость, долговременная стабильность, стабильность к внешним неблагоприятным факторам при функционировании и т. д. зависят от физических принципов работы приборов, применяемых материалов при производстве ИМС и параметров технологических процессов при их изготовлении.

На развитие микропроцессорной техники решающее значение оказывает технология производства интегральных схем.

Полупроводниковые интегральные микросхемы подразделяются на биполярные ИМС- и МОП-схемы, причем первые – более быстродействующие, а вторые имеют большую степень интеграции, меньшую потребляемую мощность и меньшую стоимость. Цифровые микросхемы могут по идеологии, конструкторскому решению, технологии относиться к разным семействам, но выполнять одинаковую функцию, т. е. быть инвертором, триггером или процессором. Наиболее популярными семействами можно назвать у биполярных ИМС: ТТЛ (транзисторно-транзисторная логика), ТТЛШ (с диодами Шоттки), ЭСЛ (эмиттерно-связанная логика); у МДП: n-МОП и КМОП.

Базовым материалом для изготовления ИМС является кремний. Несмотря на то, что он не обладает высокой подвижностью носителей заряда, а значит приборы на его основе теоретически будут уступать по быстродействию приборам на основе арсенида галлия GaAs, однако система Si-SiO₂ существенно более технологична. С другой стороны, приборы на кремниевой основе (кремний – оксид кремния) обладают совершенной границей раздела Si-SiO₂, химической стойкостью, электрической прочностью и другими уникальными свойствами.

Технологический цикл производства ИМС включает:

- эпитаксиальное наращивание слоя на подготовленную подложку;
- наращивание слоя SiO₂ на эпитаксиальный слой;
- нанесение фоторезиста, маскирование и вытравливание окон в слое;
- легирование примесью путем диффузии или имплантацией;
- аналогично повторение операций для подготовки других легированных областей;
- повторение операций для создания окон под контактные площадки;
- металлизацию всей поверхности алюминием или поликремнием;
- повторение операций для создания межсоединений;
- удаление излишков алюминия или поликремния;
- контроль функционирования;
- помещение в корпус;
- выходной контроль.

Наиболее критичным для увеличения степени интеграции является процесс литографии, т. е. процесс переноса геометрического рисунка шаблона на поверхность кремниевой пластины. С помощью этого рисунка формируют такие элементы схемы, как электроды затвора, контактные окна, металлические межкомпонентные соединения и т. п. На первой стадии изготовления ИМС после завершения испытаний схемы или моделирования с помощью ЭВМ формируют геометрический рисунок топологии схемы. С помощью электронно-лучевого устройства или засветки другим способом топологический рисунок схемы последовательно (уровень за уровнем) можно переносить непосредственно на поверхность кремниевой пластины, но чаще на фоточувствительные стеклянные пластины, называемые фотошаблонами. Между переносом топологического рисунка с двух шаблонов могут быть проведены операции ионной имплантации, загонки, окисления и металлизации. После экспонирования пластины помещают в раствор, который проявляет изображение в фоточувствительном материале – фоторезисте.

Увеличивая частоту колебаний световой волны, можно уменьшить ширину линии рисунка, т. е. сократить размеры интегральных схем. Но возможности этой технологии ограничены, поскольку рентгеновские лучи трудно сфокусировать. Один из вариантов – использовать сам свет в качестве шаблона (так называемое позиционирование атомов сфокусированным лазерным лучом). Этим способом, осветив двумя взаимно перпендикулярными лазерными пучками, можно изготовить решетку на кремниевой пластине из хромированных точек размером 80 нм. Сканируя лазером поверхность для создания произвольного рисунка интегральных наносхем, теоретически можно создавать схемы с шириной линии рисунка в 10 раз меньшей, чем сегодняшние. Второе ограничение при литографии накладывает органическая природа фоторезиста. Путь ее решения – применение неорганических материалов, например оксидов ванадия.

Физические процессы, протекающие в изделиях микроэлектроники (и в микросхемах памяти тоже), технология изготовления и конструктивные особенности ИМС высокой степени интеграции могут влиять на архитектуру и методы проектирования ЭВМ и систем. Естественно, уменьшение геометрических размеров транзисторов приводит к увеличению электрических полей, особенно в районе стока. Это может привести к развитию лавинного пробоя и, как следствие, к изменению выходной ВАХ МОП-транзистора:

- включению паразитного биполярного транзистора (исток–подложка–сток);
- неравномерному зарядению диэлектрика у стока;
- деградации приповерхностной области полупроводника;
- пробоем диэлектрика.

Поэтому необходимо уменьшение напряжения питания СБИС до 3,6, 3,3, 3 В и т. п. При этом известно, что блок питания компьютера обеспечивает обычно напряжения +5В, +12В, –12В.

Однако инжекция и зарядение диэлектрика – не всегда процесс отрицательный или паразитный. Уменьшение напряжения записи информационного заряда в репрограммируемых ЗУ ниже 12 В позволяет их программировать внутри микропроцессорной системы, а не специальным устройством (программатором). Тогда для разработчика открываются большие возможности для программирования не только адреса микросхем контроллера или адаптера в пространстве устройств ввода/вывода или номера прерывания, но и творить необходимое устройство самому (если иметь такую ИМС). Однако отметим, что кроме "хозяина" это может сделать и компьютерный вирус, который будет, естественно, разрушать, а не созидать что-либо.

4.1. Основные характеристики полупроводниковой памяти

Полупроводниковая память имеет большое число характеристик и параметров, которые необходимо учитывать при проектировании систем [2, 5, 6, 9]:

1. Емкость памяти определяется числом бит хранимой информации. Емкость кристалла обычно выражается также в битах и составляет: 1024 бита, 4 кбит, 16 кбит, 64 кбит и т. п. Важной характеристикой кристалла является информационная организация кристалла памяти $M \times N$, где M – число слов, N – разрядность слова. Например, кристалл емкостью 16 кбит может иметь различную организацию: 16 кбит \times 1, 4 кбит \times 4, 2 кбит \times 8. При одинаковом времени обращения память с большей шириной выборки обладает большей информационной емкостью.

2. Временные характеристики памяти.

Время доступа – временной интервал, определяемый от момента, когда центральный процессор выставил на шину адреса адрес требуемой ячейки памяти и послал по шине управления приказ на чтение или запись данных, до момента осуществления связи адресуемой ячейки с шиной данных.

Время восстановления – это время, необходимое для приведения памяти в исходное состояние после того, как ЦП снял с ША – адреса, с ШУ – сигнал "чтение" или "запись" и с ШД – данные.

3. Удельная стоимость запоминающего устройства определяется отношением его стоимости к информационной емкости, т. е. определяется стоимостью бита хранимой информации.

4. Потребляемая энергия (или рассеиваемая мощность) приводится для двух режимов работы кристалла: режима пассивного хранения информации и активного режима, когда операции записи и считывания выполняются с номинальным быстродействием. Кристаллы динамической МОП-памяти в резервном режиме потребляют примерно в десять раз меньше энергии, чем в активном режиме. Наибольшее потребление энергии, не зависящее от режима работы, характерно для кристаллов биполярной памяти.

5. Плотность упаковки определяется площадью запоминающего элемента и зависит от числа транзисторов в схеме элемента и используемой технологии. Наибольшая плотность упаковки достигнута в кристаллах динамической МОП-памяти.

6. Допустимая температура окружающей среды обычно указывается отдельно для активной работы, для пассивного хранения информации и для нерабочего состояния с отключенным питанием.

Указывается тип корпуса, если он стандартный, или чертеж корпуса с указанием всех размеров, маркировкой и нумерацией контактов, если корпус новый. Приводятся также условия эксплуатации: рабочее положение, механические воздействия, допустимая влажность и др.

4.2. Постоянные запоминающие устройства

Программируемые постоянные запоминающие устройства (ППЗУ) делятся на однократно программируемые (например, биполярные ПЗУ с плавкими соединениями) и рассматриваемые здесь многократно электрически программируемые МОП ПЗУ. Это полевой транзистор с плавающим затвором и МДОП-транзистор (металл–диэлектрик–оксид–полупроводник). Обычно в качестве диэлектрика используют нитрид кремния [2, 5, 6, 9].

4.2.1. Полевой транзистор с плавающим затвором

Конструкция и обозначение полевого транзистора с плавающим затвором представлены на рис. 14.

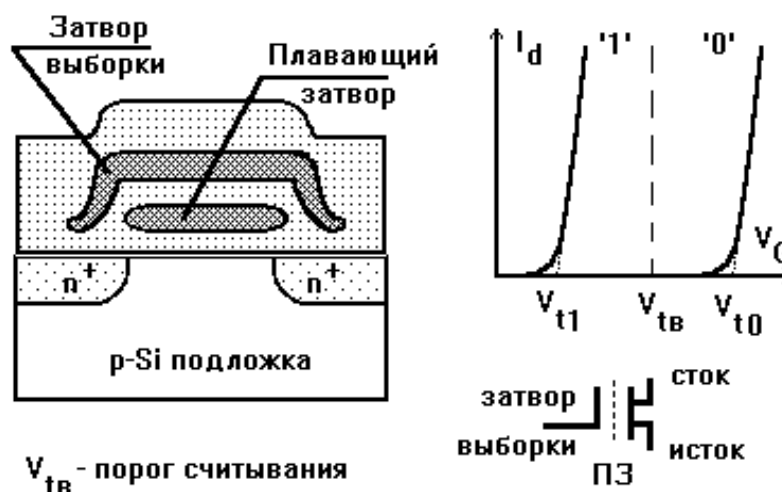


Рис. 14. МОП-транзистор с плавающим затвором

Это р-канальный нормально закрытый МОП-прибор. Здесь же показаны вольт-амперные характеристики (ВАХ) транзистора в состоянии логической единицы и нуля (до и после записи информационного заряда). Плавающий затвор представляет собой область поликремния, окруженную со всех сторон диэлектриком, т. е. он электрически не связан с другими электродами и его потенциал "плавает". Обычно толщина нижнего диэлектрического слоя составляет десятки ангстрем.

Это позволяет в сильном электрическом поле инжектировать электроны в плавающий затвор:

или сквозь потенциальный барьер Si-SiO₂ путем квантовомеханического туннелирования;

или над барьером "горячих" носителей, разогретых в поперечном или продольном поле при пробое кремниевой подложки.

Положительное смещение на верхнем затворе (относительно полупроводниковой подложки) вызовет накопление электронов в плавающем затворе при условии, что утечка электронов через верхний диэлектрический слой мала. Величина заряда Q , накопленного за время t , а значит и пороговое напряжение, определяется как

$$Q = \int J(t)dt, \quad (5)$$

где $J(t)$ – величина инжекционного тока в момент времени t .

Лавинный пробой подложки вблизи стока может приводить к неоднородной деградации транзистора и, как следствие, к ограничению по числу переключений элемента памяти. МДП-транзистор с плавающим затвором (рис. 15) может быть использован в качестве элемента памяти с временем хранения, равным времени диэлектрической релаксации структуры, которое может быть очень велико и, в основном, определяется низкими токами утечки через барьер Si-SiO₂ ($\Phi_e = 3.2$ эВ). Φ_e – высота потенциального барьера. Такой элемент памяти обеспечивает возможность непрерывного считывания без разрушения информации, причем запись и считывание могут быть выполнены в очень короткое время.

4.2

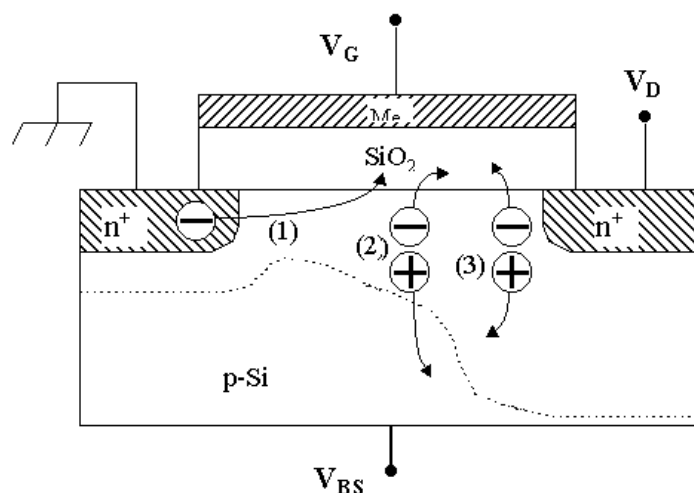


Рис. 15. Инжекция горячих электронов в диэлектрик МДП-транзистора и другие процессы, проходящие при лавинном пробое подложки

МНОП-транзистор

На рис. 16 приведена конструкция МНОП-транзистора (металл – нитрид кремния – оксид кремния – полупроводник). Эффект памяти основан на изменении порогового напряжения транзистора при наличии захваченного в подзатворном диэлектрике положительного или отрицательного заряда, который хранится на глубоких (1.3–1.5 эВ) ловушках в нитриде кремния вблизи границы $\text{SiO}_2\text{--Si}_3\text{N}_4$.

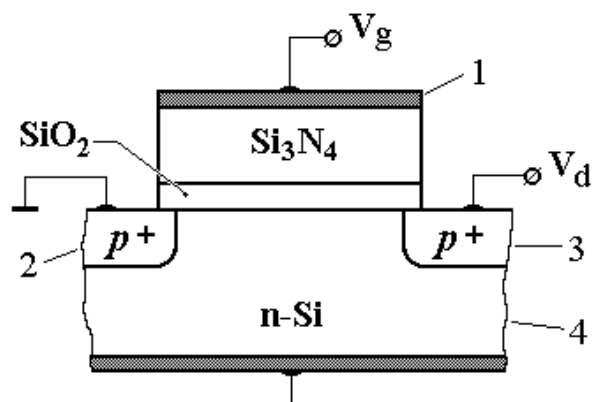


Рис. 16. Конструкция МНОП-транзистора:
1 – металлический затвор; 2,3 – области истока и стока, соответственно; 4 – подложка

Запись информационного заряда (см. рис. 17) происходит так же, как и в МОП-транзисторе с плавающим затвором. Высокая эффективность захвата электронов (или дырок) связана с большим сечением захвата на ловушки (порядка $10\text{--}13\text{ см}^2$) и с большой их концентрацией (порядка 10^{19} см^{-3}).

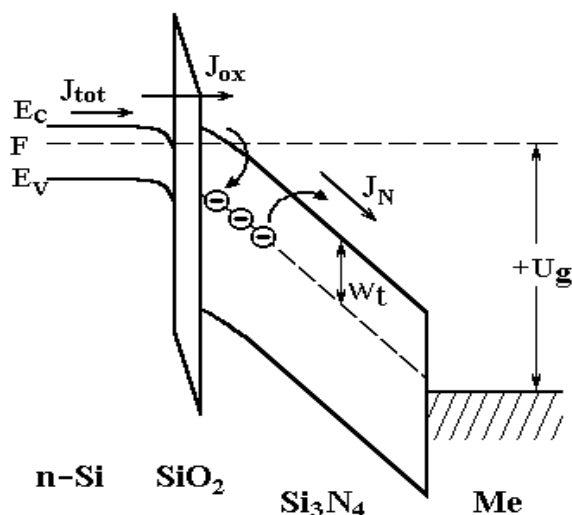


Рис. 17. Операция записи в МНОП-структуре (зонная диаграмма)

Ток в окисле J_{ox} – туннельный ток инжекции, ток J_N – ток сквозной проводимости в нитриде. В случае прямого туннелирования электронов в зону проводимости SiO_2 сквозь треугольный барьер плотность тока определяется уравнением Фаулера–Нордгейма:

$$J_{ox} = AE^2 \cdot \exp\left(-\frac{E_o}{E}\right), \quad (6)$$

где A – константы, E – напряженность электрического поля. По мере накопления заряда поле на контакте уменьшается, что приводит к уменьшению скорости записи. Эффективность записи зависит также и от тока сквозной проводимости в нитриде.

Стирание информации (возврат структуры в исходное состояние) может осуществляться:

- ультрафиолетовым излучением с энергией квантов более 5.1 эВ (ширина запрещенной зоны нитрида кремния) через кварцевое окно;
- подачей на структуру импульса напряжения, противоположного по знаку записываемому.

В соответствии с ГОСТом, такие ИМС имеют в своем названии литеры РФ и РР, соответственно. Время хранения информации в МНОП-транзисторе обусловлено термической эмиссией с глубоких ловушек и составляет порядка 10 лет в нормальных условиях. Основными факторами, влияющими на запись и хранение заряда, являются электрическое поле, температура и радиация. Количество электрических циклов *запись-стирание* обычно не менее 10^5 .

4.3. Оперативные запоминающие устройства

Полупроводниковые запоминающие устройства подразделяются на ЗУ с произвольной выборкой и ЗУ с последовательным доступом. ЗУПВ подразделяются на [5, 6, 9]:

- статические оперативные запоминающие устройства (СОЗУ);
- динамические оперативные запоминающие устройства (ДОЗУ).

ЗУ с последовательным доступом подразделяются на:

- регистры сдвига;
- приборы с зарядовой связью (ПЗС).

В основе большинства современных ОЗУ лежат комплиментарные МОП (КМОП) ИМС, которые отличаются малой потребляемой мощностью. Это достигается применением пары МОП-транзисторов с разным типом канала: n-МОП и p-МОП. Как видно на рис. 18, в КМОП-инверторе как при низком, так и при высоком уровне сигнала на входе

один из транзисторов закрыт. Поэтому потребление энергии происходит только при переключении "1" в "0" (и обратно).

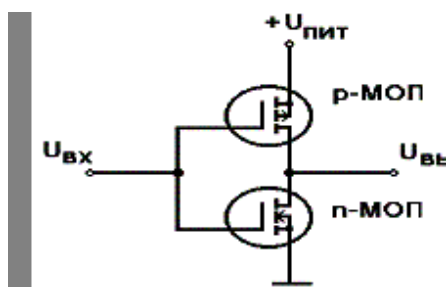


Рис. 18. Схема КМОП-инвертора

Чтобы реализовать на подложке n-типа не только p-канальный транзистор, но и n-канальный, последний изготавливается в так называемом *кармане*, как показано на рис. 19.

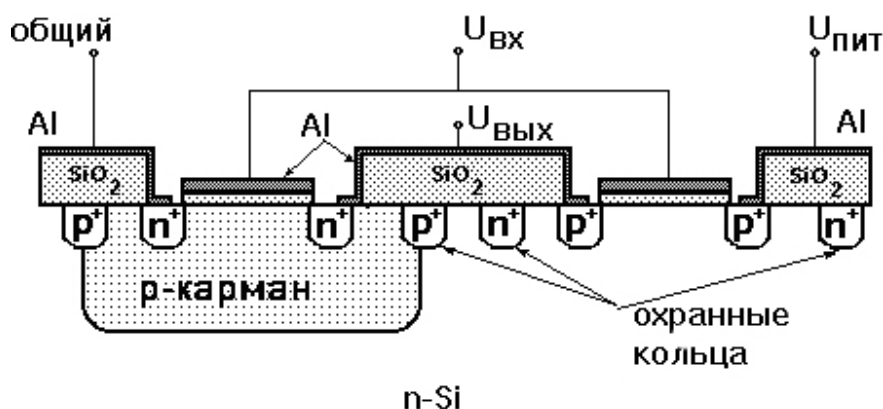


Рис. 19. Конструкция инвертора на КМОП-транзисторах

Аналогично на четырех МОП-транзисторах (2 n-МОП и 2 p-МОП, включенных параллельно и последовательно) можно построить и другие базовые логические элементы "И" и "ИЛИ" и, соответственно, на их основе строятся все другие более сложные логические схемы.

Как известно, быстродействие МОП-транзисторов в первую очередь ограничивается большой входной емкостью затвор-исток (подложка). Уменьшение геометрических размеров приборов (площади затвора и длины канала) при увеличении степени интеграции увеличивает граничную частоту.

Малое потребление энергии позволяет использовать КМОП интегральные микросхемы с питанием от микробатареи как ПЗУ, где располагается часть операционной системы, которая осуществляет начальную загрузку всей системы (программа Setup).

4.3.1. Статические запоминающие устройства

Элементарной ячейкой статического ОЗУ с произвольной выборкой является триггер на транзисторах Т1–Т4 (рис. 20) с ключами Т5–Т8 для доступа к шине данных. Причем Т1–Т2 – это нагрузки, а Т3–Т4 – нормально закрытые элементы [5, 6, 9].

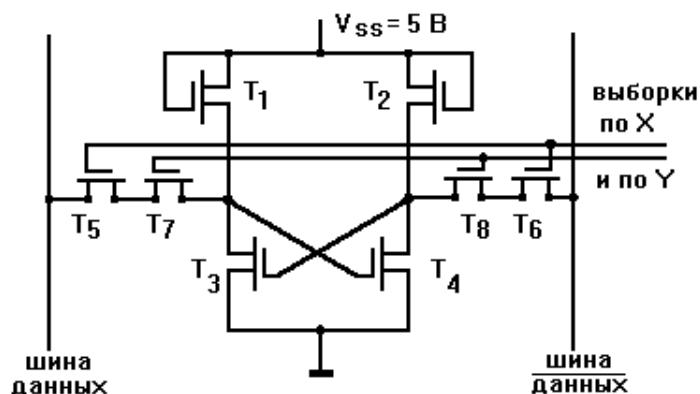


Рис. 20. Ячейка статического ОЗУ

Сопротивление элементов Т1–Т2 легко регулируется в процессе изготовления транзистора путем подгонки порогового напряжения при легировании поликремниевого затвора методом ионной имплантации. Количество транзисторов (6 или 8) на ячейку зависит от логической организации памяти микропроцессорной системы.

4.3.2. Динамические запоминающие устройства

В отличие от статических ЗУ, которые хранят информацию пока включено питание, в динамических ЗУ необходима постоянная регенерация информации. Однако при этом для хранения одного бита в ДОЗУ нужны всего 1–2 транзистора и накопительный конденсатор (рис. 21) [5, 6, 9]. Такие схемы более компактны.

Естественно, что в микросхеме динамического ОЗУ есть один или несколько тактовых генераторов и логическая схема для восстановления информационного заряда, стекающего с конденсатора. Это несколько "утяжеляет" конструкцию ИМС.



Рис. 21. Запоминающая ячейка динамического ОЗУ

Чаще всего и СОЗУ, и ДОЗУ выполнены в виде ЗУ с произвольной выборкой, которые имеют ряд преимуществ перед ЗУ с последовательным доступом.

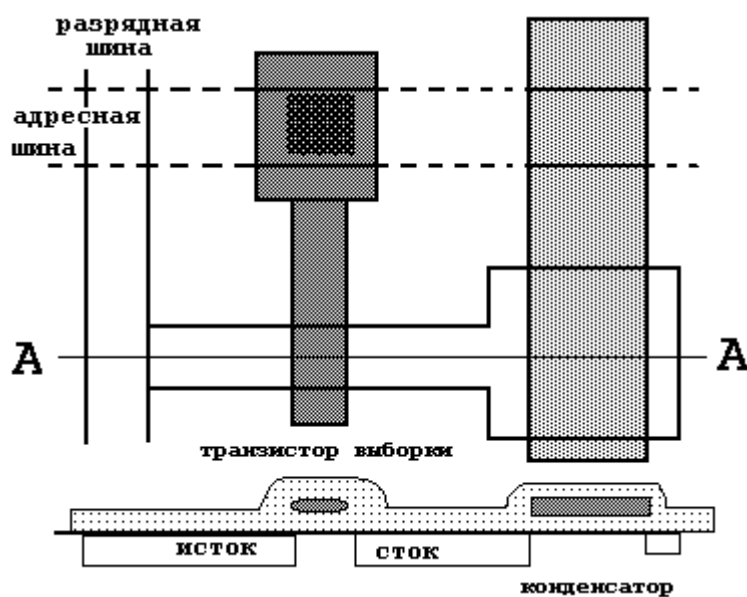


Рис. 22. Конструкция ячейки ДОЗУ (см. рис. 21, слева).
(Снизу представлен разрез схемы по линии А-А)

4.4. Запоминающие устройства с произвольной выборкой

На рис. 23 показано обозначение запоминающее устройство с произвольной выборкой (ЗУПВ) и его внутренняя структура [2, 9].

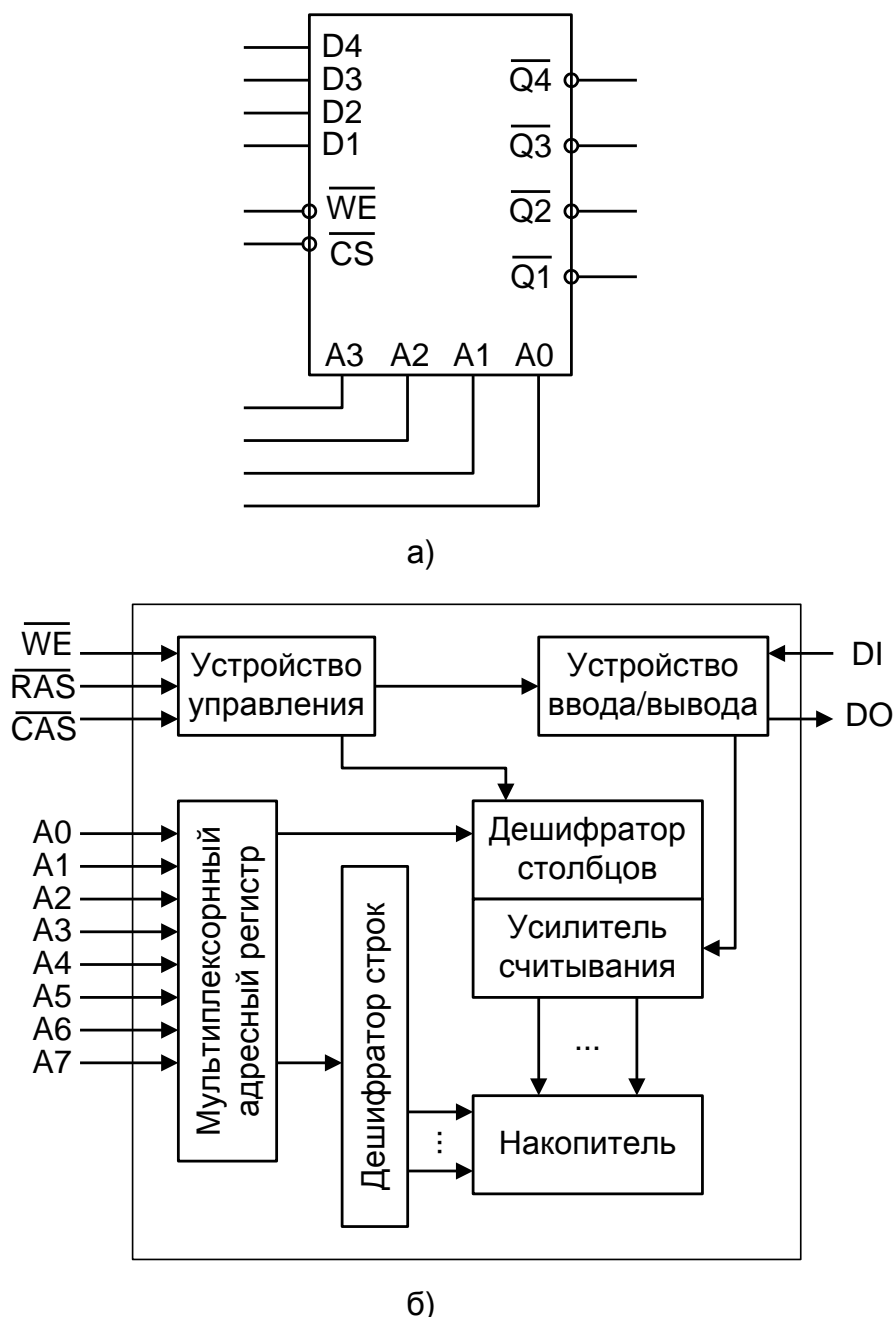


Рис. 23. ОЗУ с произвольной выборкой: а) ОЗУ 16x4 бит; б) ОЗУ 64 кбит x1 (КР565РУ5)

Здесь: D1–Dn (справа DI) – информационные входы; Q1–Qn (справа DO) – инверсные выходы; A1–An – адресные входы; WE# – запись/чтение; CS# (Chip Select) – выбор кристалла; CAS# (Column

Address Strobe) и RAS# (Row Address Strobe) – сигнал выборки столбца и строки соответственно.

Представленное здесь ЗУПВ – это ДОЗУ с организацией хранения информации 65536 бит на 1 разряд. Накопительная матрица с однотранзисторными запоминающими элементами имеет размер 512x128. Для уменьшения количества задействованных ножек у ИМС (16-входовый DIP-корпус) применена мультипликация адреса, что видно на рисунке по наличию отдельных дешифраторов строк и столбцов. Устройство управления включает два генератора тактовых сигналов и генератор сигналов записи и обеспечивает 4 режима работы: записи, считывания, регенерации и мультипликации адреса. Время регенерации – 2 мс.

Для сравнения на рис. 24 показана конструкция ППЗУ с ультрафиолетовым (УФ) стиранием на МНОП-структурах с организацией представления информации 2 кб x 8 (16384 бита).

4.5. Микросхемы памяти в составе микропроцессорной системы

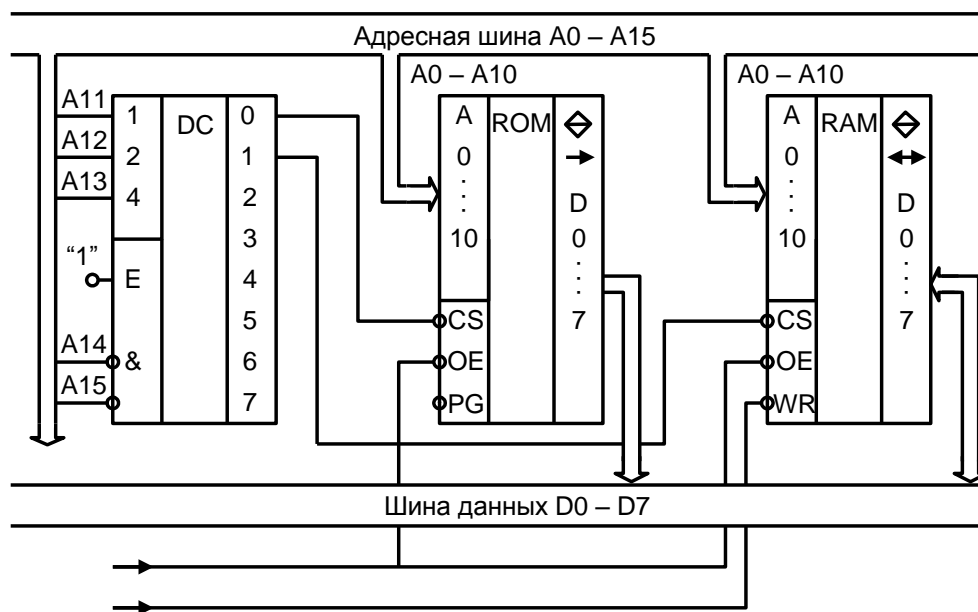


Рис. 24. Микросхемы ОЗУ (K573PY9) и ППЗУ (K573PФ5) в составе микропроцессорной системы

На рис. 24 представлено взаимодействие K573PФ2(5) и K573PY9, имеющих одинаковую организацию 2 кбит x 8, с системной магистра-

лью [9]. Байт данных с шины данных (линии D0–D7) считывается (или записывается) по адресу, выставленному на шине адреса (линии A0–A10). Естественно, число адресуемых ячеек составляет $2^{11} = 800h = 2048$. Микросхема-дешифратор K555ИД7 посредством сигнала CS# (выбор кристалла) позволяет выбрать положение ИМС ЗУ в адресном пространстве. Для данного случая это адреса 0000h–07FFh для ПЗУ(ROM) и 0800h-0FFFh для ОЗУ(RAM). Низкий уровень сигналов управления MEMW# и MEMR# активизирует процесс записи и чтения, соответственно. Напомним, что запись информации в данную ИМС ПЗУ возможен только вне микропроцессорной системы в специальном программаторе после УФ-стирания путем подачи достаточно высокого напряжения на вход PG.

4.6. Буферная память

В вычислительных системах используются подсистемы с различным быстродействием, и, в частности, с различной скоростью передачи данных (рис. 25). Обычно обмен данными между такими подсистемами реализуется с использованием прерываний или канала прямого доступа к памяти. В первую очередь, подсистема 1 формирует запрос на обслуживание по мере готовности данных к обмену. Однако обслуживание прерываний связано с непроизводительными потерями времени и при пакетном обмене производительность подсистемы 2 заметно уменьшается. При обмене данными с использованием канала прямого доступа к памяти подсистема 1 передает данные в память подсистемы 2. Данный способ обмена достаточно эффективен с точки зрения быстродействия, но для его реализации необходим довольно сложный контроллер прямого доступа к памяти.



Рис. 25. Применение буферной памяти

Наиболее эффективно обмен данными между подсистемами с различным быстродействием реализуется при наличии между ними спе-

циальной буферной памяти [2, 9]. Данные от подсистемы 1 временно запоминаются в буферной памяти до готовности подсистемы 2 принять их. Емкость буферной памяти должна быть достаточной для хранения тех блоков данных, которые подсистема 1 формирует между считываниями их подсистемой 2. Отличительной особенностью буферной памяти является запись данных с быстроедействием и под управлением подсистемы 1, а считывание – с быстроедействием и под управлением подсистемы 2 ("эластичная память"). В общем случае память должна выполнять операции записи и считывания совершенно независимо и даже одновременно, что устраняет необходимость синхронизации подсистем. Буферная память должна сохранять порядок поступления данных от подсистемы 1, т. е. работать по принципу "первое записанное слово считывается первым" (First Input First Output – FIFO). Таким образом, под буферной памятью типа FIFO понимается ЗУПВ, которое автоматически следит за порядком поступления данных и выдает их в том же порядке, допуская выполнение независимых и одновременных операций записи и считывания. На рис. 26 приведена структурная схема буферной памяти типа FIFO емкостью 64x4.

На кристалле размещены 64 4-битных регистра с независимыми цепями сдвига, организованных в 4-х последовательных 64-битных регистрах данных, 64-битный управляющий регистр, а также схема управления. Входные данные поступают на линии DI0–DI3, а вывод данных осуществляется через контакты DO0–DO3. Ввод (запись) данных производится управляющим сигналом SI (shift in), а вывод (считывание) – сигналом вывода SO (shift out). Ввод данных осуществляется только при наличии сигнала готовности ввода IR (input ready), а вывод – при наличии сигнала готовности вывода OR (output ready). Управляющий сигнал R (reset) производит сброс содержимого буфера.

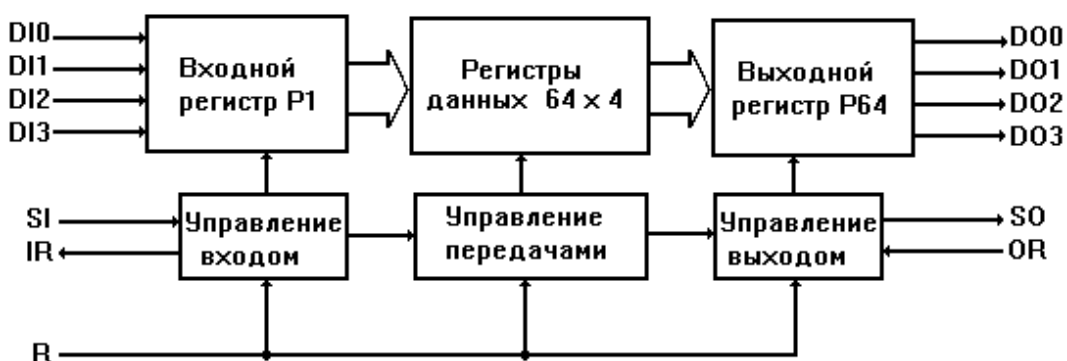


Рис. 26. Структурная схема буфера 64x4

При вводе 4-битного слова под действием сигнала SI оно автоматически передвигается в ближайший к выходу свободный регистр. Состояние регистра данных отображается в соответствующем ему управляющем триггере, совокупность триггеров образует 64-битный управляющий регистр. Если регистр содержит данные, то управляющий триггер находится в состоянии 1, а если регистр не содержит данных, то триггер находится в состоянии 0. Как только управляющий бит соседнего справа регистра изменяется на 0, слово данных автоматически сдвигается к выходу. Перед началом работы в буфер подается сигнал сброса R и все управляющие триггеры переводятся в состояние 0 (все регистры буфера свободны). На выводе IR формируется логическая 1, т. е. буфер готов воспринимать входные данные. При действии сигнала ввода SI входное слово загружается в регистр P1, а управляющий триггер этого регистра устанавливается в состояние 1: на входе IR формируется логический 0. Связи между регистрами организованы таким образом, что поступившее в P1 слово "спонтанно" копируется во всех регистрах данных FIFO и появляется на выходных линиях DO0–DO3. Теперь все 64 регистра буфера содержат одинаковые слова, управляющий триггер последнего регистра P64 находится в состоянии 1, а остальные управляющие триггеры сброшены при передаче данных в соседние справа регистры. Состояние управляющего триггера P64 выведено на линию готовности выхода OR; OR принимает значение 1, когда в триггер записывается 1. Процесс ввода может продолжаться до полного заполнения буфера; в этом случае все управляющие триггеры находятся в состоянии 1 и на линии IR сохраняется логический 0.

При подаче сигнала SO производится восприятие слова с линий DO0–DO3, управляющий триггер P64 переводится в состояние 1, на линии OR появляется логическая 1, а управляющий триггер P64 сбрасывается в 0. Затем этот процесс повторяется для остальных регистров и ноль в управляющем регистре перемещается ко входу по мере сдвига данных вправо.

В некоторых кристаллах буфера FIFO имеется дополнительная выходная линия флажка заполнения наполовину. На ней формируется сигнал 1, если число слов составляет более половины емкости буфера.

Рассмотренный принцип организации FIFO допускает выполнение записи и считывания данных независимо и одновременно. Скорость ввода определяется временным интервалом, необходимым для передачи данных из P1, а выводить данные можно с такой же скоростью. Единственным ограничением является время распространения данных через FIFO, равное времени передачи входного слова на выход незаполненного буфера FIFO. Оно равняется произведению времени внут-

ренного сдвига и числа регистра данных. В буферах FIFO, выполненных по МОП-технологии и имеющих емкость 64 слова, время распространения составляет примерно 30 мкс, а в биполярных FIFO такой же емкости – примерно 2 мкс.

Буферы можно наращивать как по числу слов, так и по их длине.

4.7. Стековая память

Стековой называют память, доступ к которой организован по принципу: последним записан – первым считан (Last Input First Output – LIFO). Использование принципа доступа к памяти на основе механизма LIFO началось с больших ЭВМ [2, 9]. Применение стековой памяти оказалось очень эффективным при построении компилирующих и интерпретирующих программ, при вычислении арифметических выражений с использованием польской инверсной записи. В малых ЭВМ она стала широко использоваться в связи с удобствами реализации процедур вызова подпрограмм и при обработке прерываний.

Принцип работы стековой памяти состоит в следующем (см. рис. 27). Когда слово А помещается в стек, оно располагается в первой свободной ячейке памяти. Следующее записываемое слово перемещает предыдущее на одну ячейку вверх и занимает его место и т. д. Запись 8-го кода (после Н) приводит к переполнению стека и потере кода А. Считывание слов из стека осуществляется в обратном порядке, начиная с кода Н, который был записан последним. Заметим, что выборка, например, кода Е невозможна до выборки кода F, что определяется механизмом обращения при записи и чтении типа LIFO. Для фиксации переполнения стека желательно формировать признак переполнения.

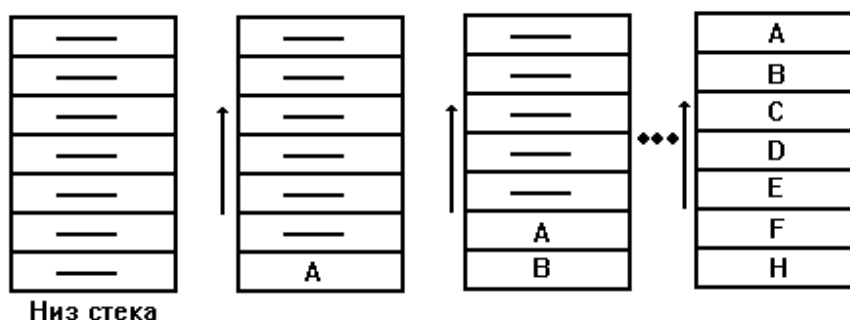


Рис. 27. Принцип работы стековой памяти

Перемещение данных при записи и считывании информации в стековой памяти подобно тому, как это имеет место в сдвигающих регистрах. С точки зрения реализации механизма доступа к стековой памяти выделяют аппаратный и аппаратно-программный (внешний) стеки.

Аппаратный стек представляет собой совокупность регистров, связи между которыми организованы таким образом, что при записи и считывании данных содержимое стека автоматически сдвигается. Обычно емкость аппаратного стека ограничена диапазоном от нескольких регистров до нескольких десятков регистров, поэтому в большинстве МП такой стек используется для хранения содержимого программного счетчика и его называют стеком команд. Основное достоинство аппаратного стека – высокое быстродействие, а недостаток – ограниченная емкость.

Наиболее распространенным в настоящее время и, возможно, лучшим вариантом организации стека в ЭВМ является использование области памяти. Для адресации стека используется указатель стека, который предварительно загружается в регистр и определяет адрес последней занятой ячейки. Помимо команд CALL и RET, по которым записывается в стек и восстанавливается содержимое программного счетчика, имеются команды PUSH и POP, которые используются для временного запоминания в стеке содержимого регистров и их восстановления, соответственно. В некоторых МП содержимое основных регистров запоминается в стеке автоматически при прерывании программ. Содержимое регистра указателя стека при записи уменьшается, а при считывании увеличивается на 1 при выполнении команд PUSH и POP, соответственно.

5. Организация ввода/вывода в микропроцессорной системе

Вводом/выводом (ВВ) называется передача данных между ядром ЭВМ, включающим в себя микропроцессор и основную память, и внешними устройствами (ВУ) [5, 6]. Это единственное средство взаимодействия ЭВМ с "внешним миром", и архитектура ВВ (режимы работы, форматы команд, особенности прерываний, скорость обмена и др.) непосредственно влияет на эффективность всей системы. За время эволюции ЭВМ подсистема ВВ претерпела наибольшие изменения благодаря расширению сферы применения ЭВМ и появлению новых внешних устройств. Особенно важную роль средства ВВ играют в управляющих ЭВМ. Разработка аппаратных средств и программного обеспечения ВВ является наиболее сложным этапом проектирования новых систем на базе ЭВМ, а возможности ВВ серийных машин представляют собой один из важных параметров, определяющих выбор машины для конкретного применения.

5.1. Программная модель внешнего устройства

Подключение внешних устройств к системной шине осуществляется посредством электронных схем, называемых контроллерами ВВ (интерфейсами ВВ). Они согласуют уровни электрических сигналов, а также преобразуют машинные данные в формат, необходимый устройству, и наоборот. Обычно контроллеры ВВ конструктивно оформляются вместе с процессором в виде интерфейсных плат.

В процессе ввода/вывода передается информация двух видов: управляющие данные (слова) и собственно данные, или данные-сообщения. Управляющие данные от процессора, называемые также командными словами или приказами, инициируют действия, не связанные непосредственно с передачей данных, например запуск устройства, запрещение прерываний и т. п. Управляющие данные от внешних устройств называются словами состояния; они содержат информацию об определенных признаках, например о готовности устройства к передаче данных, о наличии ошибок при обмене и т. п. Состояние обычно представляется в декодированной форме – один бит для каждого признака.

Регистр, содержащий группу бит, к которой процессор обращается в операциях ВВ, образует порт ВВ. Таким образом, наиболее общая программная модель внешнего устройства, которое может выполнять ввод и вывод, содержит четыре регистра ВВ: регистр выходных дан-

ных (выходной порт), регистр входных данных (входной порт), регистр управления и регистр состояния (рис. 28). Каждый из этих регистров должен иметь однозначный адрес, который идентифицируется дешифратором адреса. В зависимости от особенностей устройства общая модель конкретизируется. Например, отдельные регистры состояния и управления объединяются в один регистр, в устройстве ввода (вывода) имеется только регистр входных (выходных) данных, для ввода и вывода используется двунаправленный порт.



Рис. 28. Программная модель внешнего устройства

Непосредственные действия, связанные с вводом/выводом, реализуются одним из двух способов, различающихся адресацией регистров ВВ.

Интерфейс с изолированными шинами характеризуется отдельной адресацией памяти и внешних устройств при обмене информацией. Изолированный ВВ предполагает наличие специальных команд ввода/вывода, общий формат которых показан на рис. 29. При выполнении команды ввода IN содержимое адресуемого входного регистра PORT передается во внутренний регистр процессора – REG, а при выполнении команды OUT содержимое регистра REG передается в выходной порт PORT. В процессоре могут быть и другие команды, относящиеся к ВВ и связанные с проверкой и модификацией содержимого регистра управления и состояния.

КОП (IN)	REG	PORT
КОП (OUT)	PORT	REG

Рис. 29. Команды ввода/вывода (общий формат)

Нетрудно заметить, что в этом способе адресное пространство портов ввода и вывода изолировано от адресного пространства памяти, т. е. в ЭВМ один и тот же адрес могут иметь порт ВВ и ячейка памяти. Разделение адресных пространств осуществляется с помощью управляющих сигналов, относящихся к системам ВВ и памяти (MEMRD# – считывание данных из памяти, MEMWR# – запись данных в память, IORD# – чтение порта ВВ, IOWR# – запись в порт ВВ) (# – активный низкий уровень сигналов).

В ЭВМ, рассчитанной на изолированный ВВ, нетрудно перейти к ВВ, отображенному на память. Если, например, адресное пространство памяти составляет 64 кб (ША состоит из 16 бит), а для программного обеспечения достаточно 32 кб, то область адресов от 0 до 32 кб-1 (младшие 8 бит ША) используется для памяти, а от 32 кб до 64 кб-1 (старшие 8 бит ША) – для ввода/вывода. При этом признаком, дифференцирующим обращения к памяти и портам ВВ, может быть старший бит адреса.

Таким образом, интерфейс с общими шинами (ввод/вывод с отображением на память) имеет организацию, при которой часть общего адресного пространства отводится для внешних устройств, регистры которых адресуются так же, как и ячейки памяти. В этом случае для адресации портов ВВ используются полные адресные сигналы: READ – чтение, WRITE – запись.

В операционных системах ЭВМ имеется набор подпрограмм (драйверов ВВ), управляющих операциями ВВ стандартных внешних устройств. Благодаря им, пользователь может не знать многих особенностей ВУ и интерфейсов ВВ, а применять четкие программные протоколы.

5.2. Форматы передачи данных

Рассмотрим некоторые общие вопросы, связанные с обменом данными между ВУ и микроЭВМ. Существуют два способа передачи слов информации по линиям данных: параллельный, когда одновременно пересылаются все биты слова, и последовательный, когда биты слова пересылаются поочередно, начиная, например, с его младшего разряда.

Так как между отдельными проводниками шины для параллельной передачи данных существует электрическая емкость, то при изменении сигнала, передаваемого по одному из проводников, возникает помеха (короткий выброс напряжения) на других проводниках. С увеличением длины шины (увеличением емкости проводников) помехи возрастают и могут восприниматься приемником как сигналы. Поэтому рабочее расстояние для шины параллельной передачи данных ограничивается длиной 1–2 м, и только за счет существенного удорожания шины или снижения скорости передачи длину шины можно увеличить до 10–20 м.

Указанное обстоятельство и желание использовать для дистанционной передачи информации телеграфные и телефонные линии обусловили широкое распространение способа последовательного обмена данными между ВУ и микроЭВМ и между несколькими микроЭВМ. Возможны два режима последовательной передачи данных: синхронный и асинхронный.

При синхронной последовательной передаче каждый передаваемый бит данных сопровождается импульсом синхронизации, информирующим приемник о наличии на линии информационного бита. Следовательно, между передатчиком и приемником должны быть протянуты минимум три провода: два для передачи импульсов синхронизации и бит данных, а также общий заземленный проводник. Если же передатчик (например, микроЭВМ) и приемник (например, дисплей) разнесены на несколько метров, то каждый из сигналов (информационный и синхронизирующий) придется посылать либо по экранированному (телевизионному) кабелю, либо с помощью витой пары проводов, один из которых заземлен или передает сигнал, инверсный основному.

Синхронная последовательная передача начинается с пересылки в приемник одного или двух символов синхронизации (не путать с импульсами синхронизации). Получив такой символ (символы), приемник начинает прием данных и их преобразование в параллельный формат. Естественно, что при такой организации синхронной последовательной передачи она целесообразна лишь для пересылки массивов слов, а не

отдельных символов. Это обстоятельство, а также необходимость использования для обмена сравнительно дорогих (четырёх проводных или кабельных) линий связи помешало широкому распространению синхронной последовательности передачи данных.

Асинхронная последовательная передача данных означает, что у передатчика и приемника нет общего генератора синхроимпульсов и что синхронизирующий сигнал не посылается вместе с данными. Как же в таком случае приемник будет узнавать о моментах начала и завершения передачи бит данных? Опишем простую процедуру, которую можно использовать, если передатчик и приемник асинхронной последовательной передачи данных согласованы по формату и скорости передачи.

Стандартный формат асинхронной последовательной передачи данных, используемый в ЭВМ и ВУ, содержит n пересылаемых бит информации (при пересылке символов n равно 7 или 8 битам) и 3–4 дополнительных бита: стартовый бит, бит контроля четности (или нечетности) и 1 или 2 стоповых бита (рис. 30, а). Бит четности (или нечетности) может отсутствовать. Когда передатчик бездействует (данные не посылаются на линию), на линии сохраняется уровень сигнала, соответствующий логической 1 [5, 6].



Рис. 30. Формат асинхронной последовательной передачи данных

Передатчик может начать пересылку символа в любой момент времени посредством генерирования стартового бита, т. е. перевода линии в состояние логического 0 на время, точно равное времени передачи бита. Затем происходит передача битов символа, начиная с младшего значащего бита, за которым следует дополнительный бит контро-

ля по четности или нечетности. Далее с помощью стопового бита линия переводится в состояние логической единицы (рис. 30, б). При единичном бите контроля стоповый бит не изменяет состояния сигнала на линии. Состояние логической единицы должно поддерживаться в течение промежутка времени, равного 1 или 2 временам передачи бита.

Промежуток времени от начала стартового бита до конца стопового бита (стоповых бит) называется кадром. Сразу после стоповых бит передатчик может посылать новый стартовый бит, если имеется другой символ для передачи; в противном случае уровень логической единицы может сохраняться на протяжении всего времени, пока бездействует передатчик. Новый стартовый бит может быть послан в любой момент времени после окончания стопового бита, например, через промежуток времени, равный 0.43 или 1.5 времени передачи бита.

В линиях последовательной передачи данных передатчик и приемник должны быть согласованы по всем параметрам формата, изображенного на рис. 8, включая номинальное время передачи бита. Для этого в приемнике устанавливается генератор синхроимпульсов, частота которого должна совпадать с частотой аналогичного генератора передатчика. Кроме того, для обеспечения оптимальной защищенности сигнала от искажения, шумов и разброса частоты синхроимпульсов приемник должен считывать принимаемый бит в середине его длительности. Рассмотрим работу приемника с того момента, когда он закончил прием символа данных и перешел в режим обнаружения стартового бита следующего слова.

Если линия перешла в состояние логического нуля и находится в этом состоянии в течение времени не меньшего, чем половина временного интервала передачи бита, то приемник переводится в режим считывания бит информации. В противном случае приемник остается в режиме обнаружения, так как вероятнее всего это был не стартовый бит, а шумовая помеха. В новом режиме приемник вырабатывает сигналы считывания через интервалы, равные времени передачи бита, т. е. выполняет считывание и сохранение принимаемых бит примерно на середине их передачи. Аналогичным образом будут считаны бит контроля четности и сигнал логической единицы (стоповый бит). Если оказалось, что на месте стопового бита обнаружен сигнал логического нуля, то произошла "Ошибка кадра" и символ принят неправильно. Иначе проверяется, четно ли общее число единиц в информационных битах и бите контроля, и если оно четно, производится запись принятого символа в буфер приемника.

Передний фронт стартового бита сигнализирует о начале поступления передаваемой информации, а момент его появления служит точкой отсчета времени для считывания бит данных. Стоповый бит предоставляет время для записи принятого символа в буфер приемника и обеспечивает возможность выявления ошибки кадра. Наиболее часто ошибки кадра появляются тогда, когда приемник ошибочно синхронизирован с битом 0, который в действительности не является стартовым битом. Если передатчик бездействует (посылает сигнал логической единицы) в течение одного кадра или более, то всегда можно восстановить правильную синхронизацию. Хуже обстоит дело при рассинхронизации генераторов передатчика и приемника, когда временной интервал между сигналами считывания принимаемых битов будет меньше или больше времени передачи бита.

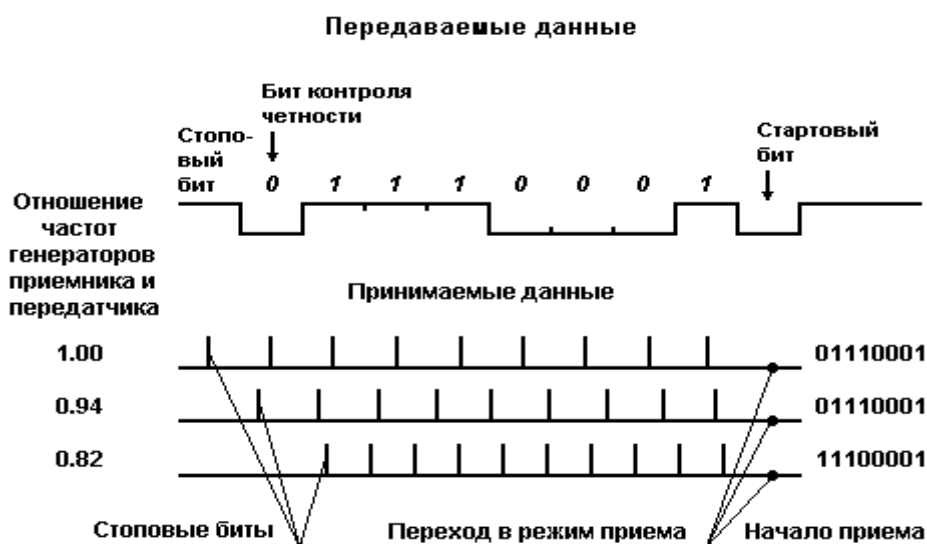


Рис. 31. Ошибка из-за рассинхронизации генераторов передатчика и приемника

Например, если при считывании битов посылки, показанной на рис. 3.3 б, временной интервал между сигналами считывания станет на 6 % меньше, чем время передачи бита, то восьмой и девятый сигналы считывания будут выработаны тогда, когда на линии находится бит контроля четности (рис. 31). Следовательно, не будет обнаружен стоповый бит и будет зафиксирована ошибка кадра, несмотря на правильность принятой информации. Однако при 18%-й рассинхронизации генераторов, когда вместо кода (01110001) приемник зафиксирует код (11100001), никаких ошибок не будет обнаружено – четность соблюдена и стоповый (девятый по порядку) бит равен 1 (см. рис. 31).

5.3. Параллельная передача данных

Параллельная передача данных между контроллером и ВУ является по своей организации наиболее простым способом обмена. Для организации параллельной передачи данных помимо шины данных, количество линий в которой равно числу одновременно передаваемых битов данных, используется минимальное количество управляющих сигналов.

В простом контроллере ВУ, обеспечивающем побайтную передачу данных на внешнее устройство (рис. 32), в шине связи с ВУ используются всего два управляющих сигнала: "Выходные данные готовы" и "Данные приняты" [2, 5, 6].

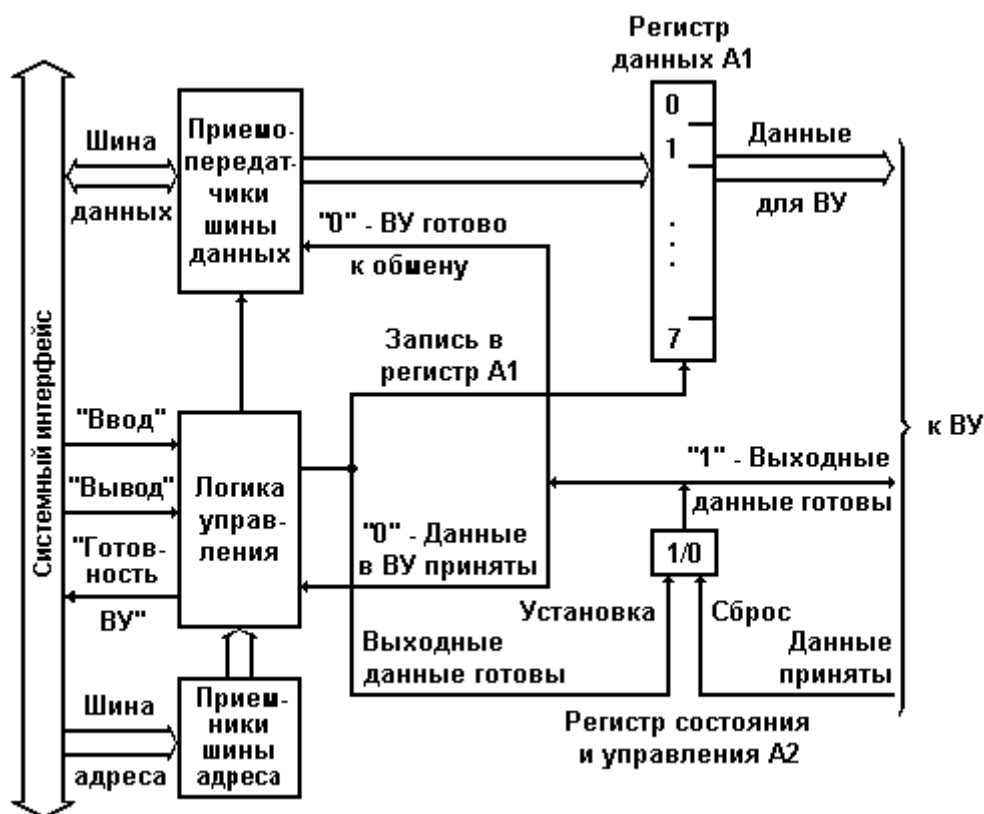


Рис. 32. Простой параллельный контроллер вывода

Для формирования управляющего сигнала "Выходные данные готовы" и приема из ВУ управляющего сигнала "Данные приняты" в контроллере используется одноразрядный адресуемый регистр состояния и управления A2 (обычно используются отдельные регистр состояния и регистр управления). Одновременно с записью очередного байта данных с шины данных системного интерфейса в адресуемый регистр

данных контроллера (порт вывода A1) в регистр состояния и управления записывается логическая единица. Тем самым формируется управляющий сигнал "Выходные данные готовы" в шине связи с ВУ.

Внешнее устройство, приняв байт данных, управляющим сигналом "Данные приняты" обнуляет регистр состояния контроллера. При этом формируется управляющий сигнал системного интерфейса "Готовность ВУ" и признак готовности ВУ к обмену, передаваемый в процессор по одной из линий шины данных системного интерфейса посредством стандартной операции ввода при реализации программы асинхронного обмена.

Логика управления контроллера обеспечивает селекцию адресов регистров контроллера, прием управляющих сигналов системного интерфейса и формирование на их основе внутренних управляющих сигналов контроллера, формирование управляющего сигнала системного интерфейса "Готовность ВУ". Для сопряжения регистров контроллера с шинами адреса и данных системного интерфейса в контроллере используются соответственно приемники шины адреса и приемопередатчики шины данных.

Рассмотрим на примере, каким образом контроллер ВУ обеспечивает параллельную передачу данных в ВУ под управлением программы асинхронного обмена. Алгоритм асинхронного обмена в данном случае передачи прост.

1. Процессор микроЭВМ проверяет готовность ВУ к приему данных.

2. Если ВУ готово к приему данных (в данном случае это логический 0 в нулевом разряде регистра A2), то данные передаются с шины данных системного интерфейса в регистр данных A1 контроллера и далее в ВУ. Иначе повторяется п. 1.

Пример 1. Фрагмент программы передачи байта данных в асинхронном режиме с использованием параллельного контроллера ВУ (рис. 32). Для написания программы асинхронной передачи воспользуемся командами процессора 8086.

	MOV DX, A2;	номер порта A2 помещаем в DX
m1:	IN AL, DX;	чтение байта из порта A2
	TEST AL, 1;	проверка нулевого состояния регистра A2
	JNS m1;	переход на метку m1 если разряд не нулевой
	MOV AL, 64;	выводимый байт данных помещается в AL
	MOV DX, A1;	номер порта A1 записываем в DX
	OUT DX, AL;	содержимое регистра AX передаем в порт A1

Команда во второй строке приводит к следующим действиям. При ее выполнении процессор по шине адреса передает в контроллер адрес A2, сопровождая его сигналом "Ввод" (IORD#; здесь и далее в скобках указаны сигналы на шине ISA). Логика управления контроллера, реагируя на эти сигналы, обеспечивает передачу в процессор содержимого регистра состояния A2 по шине данных системного интерфейса.

Команда в третьей строке приводит к следующим действиям. Процессор проверяет значение соответствующего разряда принятых данных. Нуль в этом разряде указывает на неготовность ВУ к приему данных и, следовательно, на необходимость возврата к проверке содержимого A2, т. е. процессор, выполняя три первые команды, ожидает готовности ВУ к приему данных. Единица в этом разряде подтверждает готовность ВУ и, следовательно, возможность передачи байта данных.

В седьмой строке осуществляется пересылка данных из регистра AX процессора в регистр данных контроллера A1. Процессор по шине адреса передает в контроллер адрес A1, а по шине данных – байт данных, сопровождая их сигналом "Вывод" (IOWR#). Логика управления контроллера обеспечивает запись данных с шины данных в регистр данных A1 и устанавливает в нуль бит готовности регистра состояния A2, формируя тем самым управляющий сигнал для ВУ "Выходные данные готовы". Внешнее устройство принимает байт данных и управляющим сигналом "Данные приняты" устанавливает в единицу регистр состояния A2. Далее контроллер ВУ по этому сигналу может сформировать и передать в процессор сигнал "Готовность ВУ", который в данном случае извещает процессор о приеме данных внешним устройством и разрешает процессору снять сигнал "Вывод" и тем самым завершить цикл вывода данных в команде пересылки, однако в IBM-совместимых персональных компьютерах с шиной ISA сигнал "Готовность ВУ" не формируется, а имеется сигнал IO CH RDY#, позволяющий продлить цикл обмена, если устройство недостаточно быстрое. В данном случае нет необходимости в сигнале "Готовность ВУ", т.к. шина ISA является синхронной и, следовательно, все операции выполняются по тактовым импульсам.

Блок-схема простого контроллера ВУ, обеспечивающего побайтный прием данных из ВУ, приведена на рис. 33 [5, 6]. В этом контроллере при взаимодействии с внешним устройством также используются два управляющих сигнала: "Данные от ВУ готовы" и "Данные приняты".

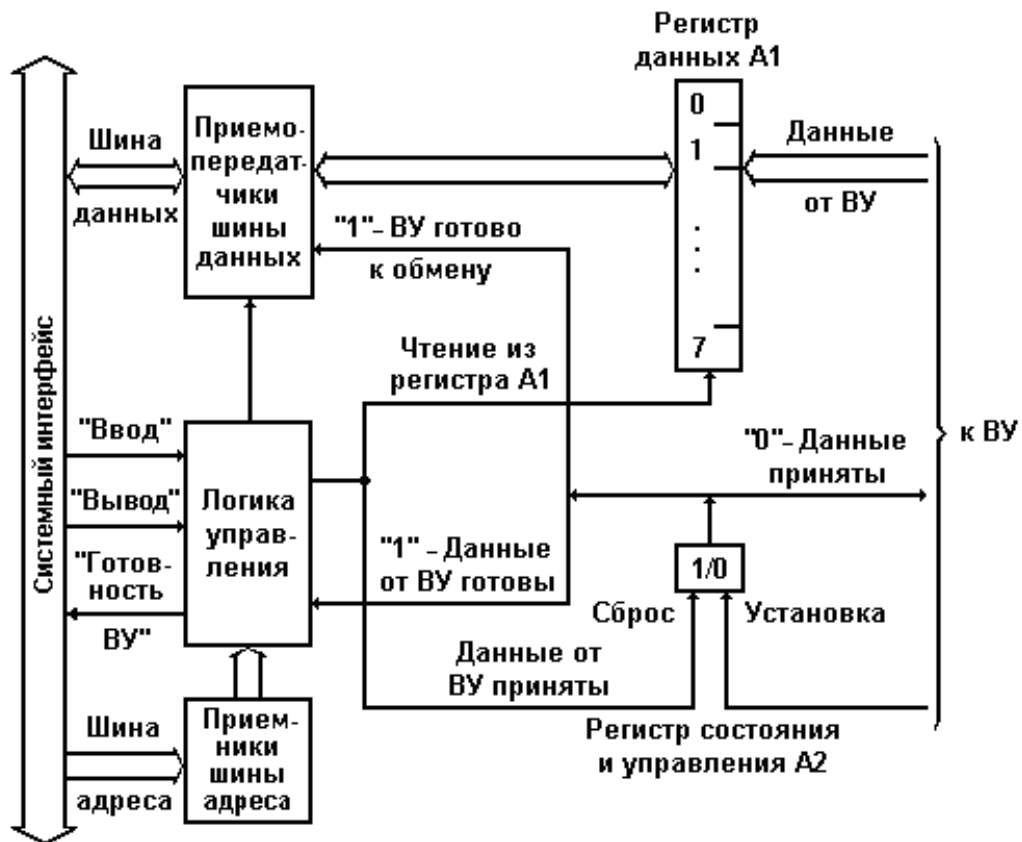


Рис. 33. Простой параллельный контроллер ввода

Для формирования управляющего сигнала "Данные приняты" и приема из ВУ управляющего сигнала "Данные от ВУ готовы" используется одноразрядный адресуемый регистр состояния и управления А2.

Внешнее устройство записывает в регистр данных контроллера А1 очередной байт данных и управляющим сигналом "Данные от ВУ готовы" устанавливает в единицу регистр состояния и управления А2.

При этом формируются: управляющий сигнал системного интерфейса "Готовность ВУ"; признак готовности ВУ к обмену, передаваемый в процессор по одной из линий шины данных системного интерфейса посредством операции ввода при реализации программы асинхронного обмена.

Тем самым контроллер извещает процессор о готовности данных в регистре А1. Процессор, выполняя программу асинхронного обмена, читает байт данных из регистра данных контроллера и обнуляет регистр состояния и управления А2. При этом формируется управляющий сигнал "Данные приняты" в шине связи с внешним устройством.

Логика управления контроллера и приемопередатчики шин системного интерфейса выполняют те же функции, что и в контроллере вывода (см. рис. 32).

Рассмотрим работу параллельного интерфейса ввода при реализации программы асинхронного обмена. Алгоритм асинхронного ввода так же прост, как и асинхронного вывода.

1. Процессор проверяет наличие данных в регистре данных контроллера A1.

2. Если данные готовы (логическая 1 в регистре A2), то они передаются из регистра данных A1 на шину данных системного интерфейса и далее в регистр процессора или ячейку памяти микрокомпьютера. Иначе повторяется п. 1.

Пример 2. Фрагмент программы приема байта данных в асинхронном режиме с использованием параллельного интерфейса (контроллер ВУ, рис. 33):

	MOV DX, A2;	номер порта A2 помещаем в DX
m1:	IN AL, DX;	чтение байта из порта A2
	TEST AL, 1;	проверка нулевого разряда состояния регистра A2
	JZ ml;	переход на метку ml если разряд не нулевой
	MOV DX, A1;	номер порта A1 записываем в DX
	IN AL, DX;	содержимое регистра A1 передаем в регистр AL

В третьей строке выполняется проверка содержимого регистра A2, т. е. признака наличия данных в регистре данных A1. Команда выполняется точно так же, как и в примере 1. Единица в нулевом разряде (содержимое регистра A2) подтверждает, что данные от ВУ записаны в регистр данных контроллера и необходимо переслать их на шину данных. Нуль в знаковом разряде указывает на неготовность данных от ВУ и, следовательно, на необходимость вернуться к проверке готовности.

IN AL, DX – пересылка данных из регистра данных контроллера A1 в регистр процессора AL. Процессор передает в контроллер по шине адреса системного интерфейса адрес A1, сопровождая его сигналом "Ввод". Логика управления контроллера в ответ на сигнал "Ввод" (IORD#) обеспечивает передачу байта данных из регистра данных A1 на шину данных и, в общем случае, но не в IBM-совместимом персональном компьютере с шиной ISA, сопровождает его сигналом "Готовность ВУ", который подтверждает наличие данных от ВУ на шине данных и по которому процессор считывает байт с шины данных и помещает его в указанный регистр. В IBM-совместимом персональном компьютере с шиной ISA процессор считывает байт с шины данных по истечении определенного времени после установки сигнала IORD#. За-

тем логика управления обнуляет регистр состояния и управления A2, формируя тем самым управляющий сигнал для внешнего устройства "Данные приняты". Таким образом завершается цикл ввода данных.

Как видно из рассмотренных примеров, для приема или передачи одного байта данных процессору необходимо выполнить всего несколько команд, время выполнения которых и определяет максимально достижимую скорость обмена данными при параллельной передаче. Таким образом, при параллельной передаче обеспечивается довольно высокая скорость обмена данными, которая ограничивается только быстродействием ВУ.

5.4. Последовательная передача данных

Использование последовательных линий связи для обмена данными с внешними устройствами возлагает на контроллеры ВУ дополнительные, по сравнению с контроллерами для параллельного обмена, функции. Во-первых, возникает необходимость преобразования формата данных: из параллельного формата, в котором они поступают в контроллер ВУ из системного интерфейса микроЭВМ, в последовательный при передаче в ВУ и из последовательного в параллельный при приеме данных из ВУ. Во-вторых, требуется реализовать соответствующий режиму работы внешнего устройства способ обмена данными: синхронный или асинхронный.

5.4.1. Синхронный последовательный интерфейс

Простой контроллер для синхронной передачи данных в ВУ по последовательной линии связи (последовательный интерфейс) представлен на рис. 34 [2, 5, 6].

Восьмиразрядный адресуемый буферный регистр контроллера A1 служит для временного хранения байта данных до его загрузки в сдвиговый регистр. Запись байта данных в буферный регистр с шины данных системного интерфейса производится так же, как и в параллельном интерфейсе (см. разд. 5.3 и рис. 32), только при наличии единицы в одноразрядном адресуемом регистре состояния контроллера A2. Единица в регистре состояния указывает на готовность контроллера принять очередной байт в буферный регистр. Содержимое регистра A2 передается в процессор по одной из линий шины данных системного интерфейса и используется для формирования управляющего сигнала системного интерфейса "Готовность ВУ". При записи очередного байта в буферный регистр A1 обнуляется регистр состояния A2.

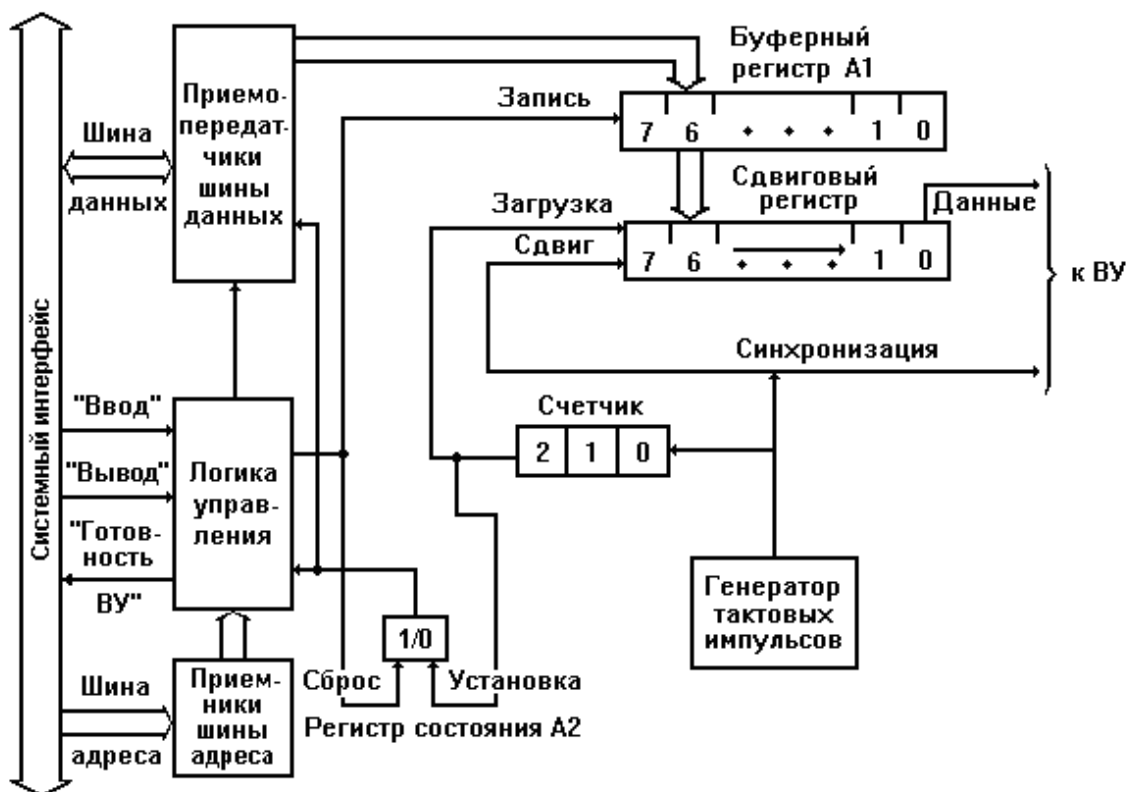


Рис. 34. Контроллер последовательной синхронной передачи

Программа записи байта данных в буферный регистр аналогична программе из примера 1 (см. разд. 5.3) за исключением команды перехода: вместо команды JNZ m1 (переход, если не нуль) необходимо использовать команду JZ m1 (переход, если нуль).

Преобразование данных из параллельного формата, в котором они поступили в буферный регистр контроллера из системного интерфейса, в последовательный и передача их на линию связи производятся в сдвиговом регистре с помощью генератора тактовых импульсов и двоичного трехразрядного счетчика импульсов следующим образом.

Последовательная линия связи контроллера с ВУ подключается к выходу младшего разряда сдвигового регистра. По очередному тактовому импульсу содержимое сдвигового регистра сдвигается на один разряд вправо и в линию связи "Данные" выдается значение очередного разряда. Одновременно со сдвигом в ВУ передается по отдельной линии "Синхронизация" тактовый импульс. Таким образом, каждый передаваемый по линии "Данные" бит информации сопровождается синхронизирующим сигналом по линии "Синхронизация", что обеспе-

чивает его однозначное восприятие на приемном конце последовательной линии связи.

Количество переданных в линию тактовых сигналов, а следовательно, и переданных бит информации подсчитывается счетчиком тактовых импульсов. Как только содержимое счетчика становится равным 7, т. е. в линию переданы 8 бит (1 байт) информации, формируется управляющий сигнал "Загрузка", обеспечивающий запись в сдвиговый регистр очередного байта из буферного регистра. Этим же управляющим сигналом устанавливается в "1" регистр состояния. Очередным тактовым импульсом счетчик будет сброшен в "0", и начнется очередной цикл выдачи восьми битов информации из сдвигового регистра в линию связи.

Синхронная последовательная передача отдельных битов данных на линию связи должна производиться без какого-либо перерыва, и следующий байт данных должен быть загружен в буферный регистр из системного интерфейса за время, не превышающее времени передачи восьми битов в последовательную линию связи.

При записи байта данных в буферный регистр обнуляется регистр состояния контроллера. Нуль в этом регистре указывает, что в линию связи передается байт данных из сдвигового регистра, а следующий передаваемый байт данных загружен в сдвиговый регистр.

Контроллер для последовательного синхронного приема данных из ВУ состоит из тех же компонентов, что и контроллер для синхронной последовательной передачи, за исключением генератора тактовых импульсов.

5.4.2. Асинхронный последовательный интерфейс

Организация асинхронного последовательного обмена данными с внешним устройством осложняется тем, что на передающей и приемной стороне последовательной линии связи используются настроенные на одну частоту, но физически разные генераторы тактовых импульсов, и следовательно общая синхронизация отсутствует. Рассмотрим на примерах организацию контроллеров последовательных интерфейсов для последовательных асинхронных передачи и приема данных.

Простейший контроллер для асинхронной передачи данных в ВУ по последовательной линии связи представлен на рис. 35 [2, 5, 6]. Он предназначен для передачи данных в формате с двумя стоповыми битами.

После передачи очередного байта данных в регистр состояния A2 записывается 1. Единичный выходной сигнал регистра A2 информирует процессор о готовности контроллера к приему следующего байта

данных и передаче его по линии связи в ВУ. Этот же сигнал запрещает формирование импульсов со схемы выработки импульсов сдвига – делителя частоты сигналов тактового генератора на 16. Счетчик импульсов сдвига (счетчик по mod 10) находится в нулевом состоянии, и его единичный выходной сигнал поступает на вентиль И, подготавливая цепь выработки сигнала загрузки сдвигового регистра.

Процесс передачи байта данных начинается с того, что процессор, выполняя команду "Вывод", выставляет этот байт на шине данных. Одновременно процессор формирует управляющий сигнал системного интерфейса "Вывод", по которому производится запись передаваемого байта в буферный регистр А1, сброс регистра состояния А2 и формирование на вентиле И сигнала "Загрузка". Передаваемый байт переписывается в разряды 1, ..., 8 сдвигового регистра, в нулевой разряд сдвигового регистра записывается 0 (стартовый бит), а в разряды 9 и 10 – 1 (стоповые биты). Кроме того, снимается сигнал "Сброс" с делителя частоты, он начинает накапливать импульсы генератора тактовой частоты и в момент приема шестнадцатого тактового импульса вырабатывает импульс сдвига.

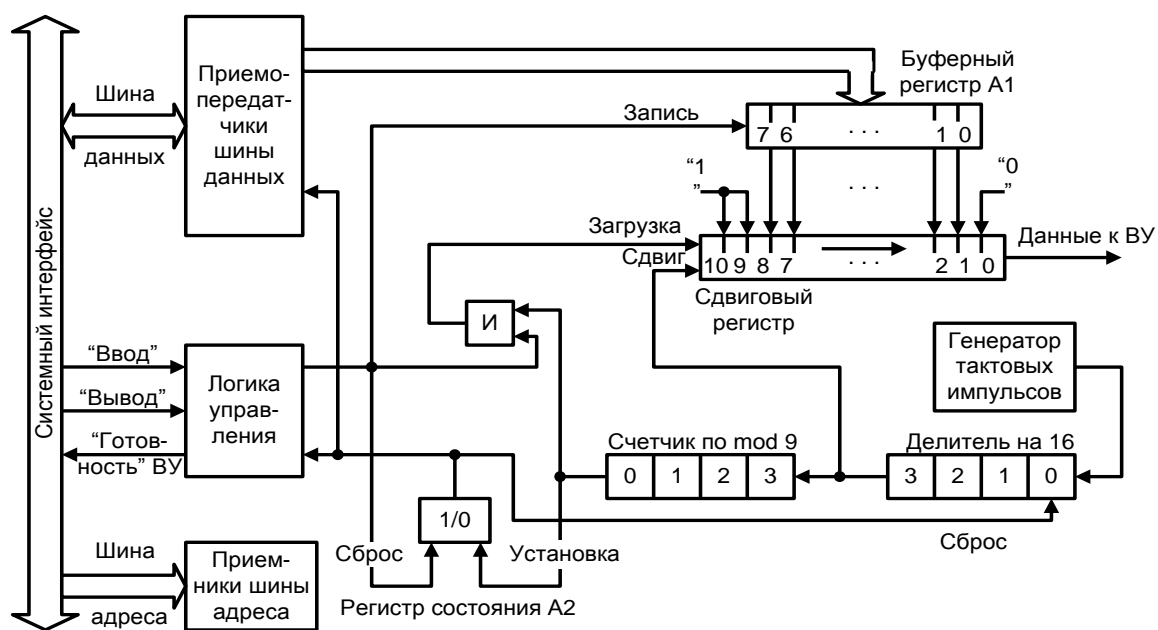


Рис. 35. Контроллер последовательной асинхронной передачи

На выходной линии контроллера "Данные" поддерживается состояние 0 (значение стартового бита) до тех пор, пока не будет выработан первый импульс сдвига. Импульс сдвига изменит состояние счетчика импульсов сдвига и переписет в нулевой разряд сдвигового регистра первый информационный бит передаваемого байта данных. Со-

стояние, соответствующее значению этого бита, будет поддерживаться на линии "Данные" до следующего импульса сдвига.

Аналогично будут переданы остальные информационные биты, первый стоповый бит и, наконец, второй стоповый бит, при передаче которого счетчик импульсов сдвига снова установится в нулевое состояние. Это приведет к записи 1 в регистр состояния A2. Единичный сигнал с выхода регистра A2 запретит формирование импульсов сдвига, а также информирует процессор о готовности к приему нового байта данных. После завершения передачи очередного кадра (стартового бита, информационного байта и двух стоповых бит) контроллер поддерживает в линии связи уровень логической единицы (значение второго стопового бита).

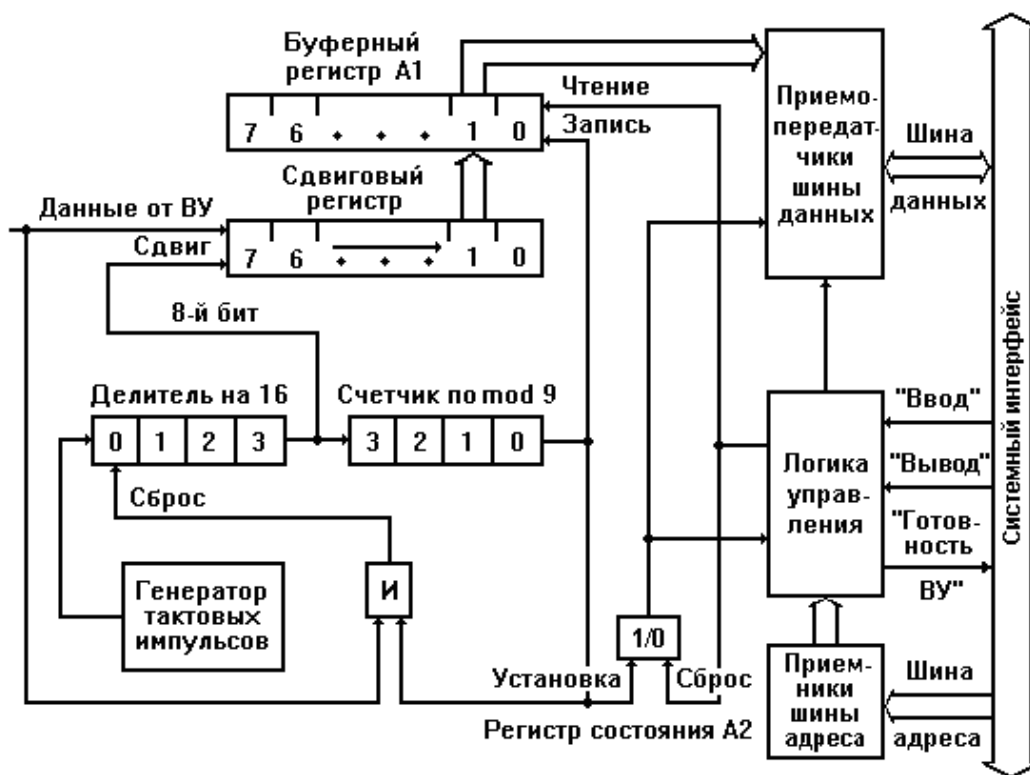


Рис. 36. Контроллер последовательного асинхронного приема

Уровень логической единицы поступает по линии "Данные" в контроллер для асинхронного приема данных (рис. 36). Этот уровень создает условия для выработки сигнала, запрещающего работу делителя частоты генератора тактовых импульсов. Действительно, после приема предыдущего байта данных счетчик импульсов сдвига (счетчик по mod 9) находится в нулевом состоянии и на вентиль И поступают два единичных сигнала: со счетчика сдвигов и из линии "Данные". На выходе

вентиля И вырабатывается сигнал сброса делителя частоты сигналов тактового генератора, запрещающий формирование импульсов сдвига.

В момент смены стопового бита на стартовый бит (момент начала передачи нового кадра) на линии "Данные" появится уровень логического нуля и тем самым будет снят сигнал сброса с делителя частоты. Состояние 4-разрядного двоичного счетчика (делителя частоты) начнет изменяться. Когда на счетчике накопится значение 8, он выдаст сигнал, поступающий на входы сдвигового регистра и счетчика импульсов сдвига. Так как частота сигналов генератора тактовых импульсов приемника должна совпадать с частотой генератора тактовых импульсов передатчика, то сдвиг (считывание) бита произойдет примерно на середине временного интервала, отведенного на передачу бита данных, т. е. времени, необходимого для выработки шестнадцати тактовых импульсов. Это делается для уменьшения вероятности ошибки из-за возможного различия частот генераторов передатчика и приемника, искажения формы передаваемых сигналов (переходные процессы) и т. п. Следующий сдвиг произойдет после прохождения шестнадцати тактовых импульсов, т. е. на середине временного интервала передачи первого информационного бита.

При приеме в сдвиговый регистр девятого бита кадра (восьмого информационного бита) из него "выдвинется" стартовый бит и, следовательно, в сдвиговом регистре будет размещен весь принятый байт информации. В этот момент счетчик импульсов сдвига придет в нулевое состояние и на его выходе будет выработан единичный сигнал, по которому содержимое сдвигового регистра переписывается в буферный регистр, в регистр состояния A2 запишется 1 и он будет информировать процессор об окончании приема очередного байта, клапан И подготовится к выработке сигнала "Сброс" (этот сигнал сформируется после прихода первого стопового бита).

Получив сигнал готовности (1 в регистре A2), процессор выполнит команду "Ввод" (см. пример 2 в разд. 5.3). При этом вырабатывается управляющий сигнал системного интерфейса "Ввод", по которому производится пересылка принятого байта данных из буферного регистра в процессор (сигнал "Чтение") и сброс регистра состояния A2.

Отметим, что для простоты изложения в контроллере на рис. 36 не показаны схемы контроля стоповых бит принимаемого кадра. Не показаны также схемы контроля четности или нечетности (паритета) передаваемой информации (обычно в передаваемом байте восьмому биту придается значение 0 или 1, так чтобы в этом байте было четное количество единиц). В реальных контроллерах имеются такие схемы, и если контроллер не принимает из линии связи нужного количества стопо-

вых бит или вырабатывается сигнал ошибки паритета в схеме контроля четности, то принятые в текущем кадре биты данных игнорируются и контроллер ожидает поступления нового стартового бита.

Обмен данными с ВУ по последовательным линиям связи широко используется в микроЭВМ, особенно в тех случаях, когда не требуется высокой скорости обмена. Вместе с тем применение в них последовательных линий связи с ВУ обусловлено двумя важными причинами. Во-первых, последовательные линии связи просты по своей организации: два провода при симплексной и полудуплексной передаче и максимум четыре – при дуплексной. Во-вторых, в микроЭВМ используются внешние устройства, обмен с которыми необходимо вести в последовательном коде.

5.5. Способы обмена информацией в микропроцессорной системе

В ЭВМ применяются три режима ввода/вывода: программно-управляемый ВВ (называемый также программным или нефорсированным ВВ), ВВ по прерываниям (форсированный ВВ) и прямой доступ к памяти. Первый из них характеризуется тем, что инициирование и управление ВВ осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и сигнализируют только о своем состоянии, в частности, о готовности к операциям ввода/вывода. Во втором режиме ВВ иницируется не процессором, а внешним устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание. Действия, выполняемые этой подпрограммой, определяются пользователем, а непосредственными операциями ВВ управляет процессор. Наконец, в режиме прямого доступа к памяти, который используется, когда пропускной способности процессора недостаточно, действия процессора приостанавливаются, он отключается от системной шины и не участвует в передачах данных между основной памятью и быстродействующим ВУ. Заметим, что во всех вышеуказанных случаях основные действия, выполняемые на системной магистрали ЭВМ, подчиняются двум основным принципам [5, 6]:

1. В процессе взаимодействия любых двух устройств ЭВМ одно из них обязательно выполняет активную, управляющую роль и является

датчиком, второе оказывается управляемым, исполнителем. Чаще всего датчиком является процессор.

2. Другим важным принципом, заложенным в структуру интерфейса, является принцип квитирования (запроса-ответа): каждый управляющий сигнал, посланный датчиком, подтверждается сигналом исполнителя. При отсутствии ответного сигнала исполнителя в течение заданного интервала времени формируется так называемый тайм-аут, датчик фиксирует ошибку обмена и прекращает данную операцию.

5.5.1. Программно-управляемый ввод/вывод

Данный режим характеризуется тем, что все действия по вводу/выводу реализуются командами прикладной программы. Наиболее простыми эти действия оказываются для "всегда готовых" внешних устройств, например индикатора на светодиодах. При необходимости ВВ в соответствующем месте программы используются команды IN или OUT. Такая передача данных называется синхронным или безусловным ВВ.

Однако для большинства ВУ до выполнения операций ВВ надо убедиться в их готовности к обмену, т. е. ВВ является асинхронным. Общее состояние устройства характеризуется флагом готовности READY, называемым также флагом готовности/занятости (READY/BUSY). Иногда состояния готовности и занятости идентифицируются отдельными флагами READY и BUSY, входящими в слово состояния устройства.

Процессор проверяет флаг готовности с помощью одной или нескольких команд. Если флаг установлен, то иницируются собственно ввод или вывод одного или нескольких слов данных. Когда же флаг сброшен, процессор выполняет цикл из 2–3 команд с повторной проверкой флага READY до тех пор, пока устройство не будет готово к операциям ВВ (рис. 37). Данный цикл называется циклом ожидания готовности ВУ и реализуется в различных процессорах по-разному [2, 5, 6].

Основной недостаток программного ВВ связан с непроизводительными потерями времени процессора в циклах ожидания. К достоинствам следует отнести простоту его реализации, не требующей дополнительных аппаратных средств.



Рис. 37. Цикл программного ожидания готовности внешнего устройства

5.5.2. Организация прерываний в микроЭВМ

Одной из разновидностей программно-управляемого обмена данными с ВУ в микроЭВМ является обмен с прерыванием программы, отличающийся от асинхронного программно-управляемого обмена тем, что переход к выполнению команд, физически реализующих обмен данными, осуществляется с помощью специальных аппаратных средств [5, 6]. Команды обмена данными в этом случае выделяют в отдельный программный модуль – подпрограмму обработки прерывания. Задачей аппаратных средств обработки прерывания в процессоре микроЭВМ как раз и является приостановка выполнения одной программы (ее еще называют основной программой) и передача управления подпрограмме обработки прерывания. Действия, выполняемые при этом процессором, как правило, те же, что и при обращении к подпрограмме. Только при обращении к подпрограмме они инициируются командой, а при обработке прерывания – управляющим сигналом от ВУ, который называют "Запрос на прерывание" или "Требование прерывания".

Эта важная особенность обмена с прерыванием программы позволяет организовать обмен данными с ВУ в произвольные моменты времени, не зависящие от программы, выполняемой в микроЭВМ. Таким образом, появляется возможность обмена данными с ВУ в реальном масштабе времени, определяемом внешней по отношению к микроЭВМ средой. Обмен с прерыванием программы существенным обра-

зом экономит время процессора, затрачиваемое на обмен. Это происходит за счет того, что исчезает необходимость в организации программных циклов ожидания готовности ВУ (см. примеры 1 и 2, разд. 5.3), на выполнение которых тратится значительное время, особенно при обмене с медленными ВУ.

Прерывание программы по требованию ВУ не должно оказывать на прерванную программу никакого влияния, кроме увеличения времени ее выполнения за счет приостановки на время выполнения подпрограммы обработки прерывания. Поскольку для выполнения подпрограммы обработки прерывания используются различные регистры процессора (счетчик команд, регистр состояния и т. д.), то информацию, содержащуюся в них в момент прерывания, необходимо сохранить для последующего возврата в прерванную программу.

Обычно задача сохранения содержимого счетчика команд и регистра состояния процессора возлагается на аппаратные средства обработки прерывания. Сохранение содержимого других регистров процессора, используемых в подпрограмме обработки прерывания, производится непосредственно в подпрограмме. Отсюда следует достаточно очевидный факт: чем больший объем информации о прерванной программе сохраняется программным путем, тем больше время реакции микроЭВМ на сигнал прерывания, и наоборот. Предпочтительными с точки зрения повышения производительности микроЭВМ (сокращения времени выполнения подпрограмм обработки, а следовательно и основной программы) являются уменьшение числа команд, обеспечивающих сохранение информации о прерванной программе, и реализация этих функций аппаратными средствами.

Формирование сигналов прерываний – запросов ВУ на обслуживание – происходит в контроллерах соответствующих ВУ. В простейших случаях в качестве сигнала прерывания может использоваться сигнал "Готовность ВУ", поступающий из контроллера ВУ в системный интерфейс микроЭВМ. Однако такое простое решение обладает существенным недостатком – процессор не имеет возможности управлять прерываниями, т. е. разрешать или запрещать их для отдельных ВУ. В результате организация обмена данными в режиме прерывания с несколькими ВУ существенно усложняется.

Для решения этой проблемы регистр состояния и управления контроллера ВУ (рис. 38) дополняют еще одним разрядом – "Разрешение прерывания". Запись 1 или 0 в разряд "Разрешение прерывания" производится программным путем по одной из линий шины данных системного интерфейса. Управляющий сигнал системного интерфейса "Запрос на прерывание" формируется с помощью схемы совпадения толь-

ко при наличии единиц в разрядах "Готовность ВУ" и "Разрешение прерывания" регистра состояния и управления контроллера [2, 5, 6].

Аналогичным путем решается проблема управления прерываниями в микроЭВМ, в целом. Для этого в регистре состояния процессора выделяется разряд, содержимое которого определяет, разрешены или запрещены прерывания от внешних устройств. Значение этого разряда может устанавливаться программным путем.

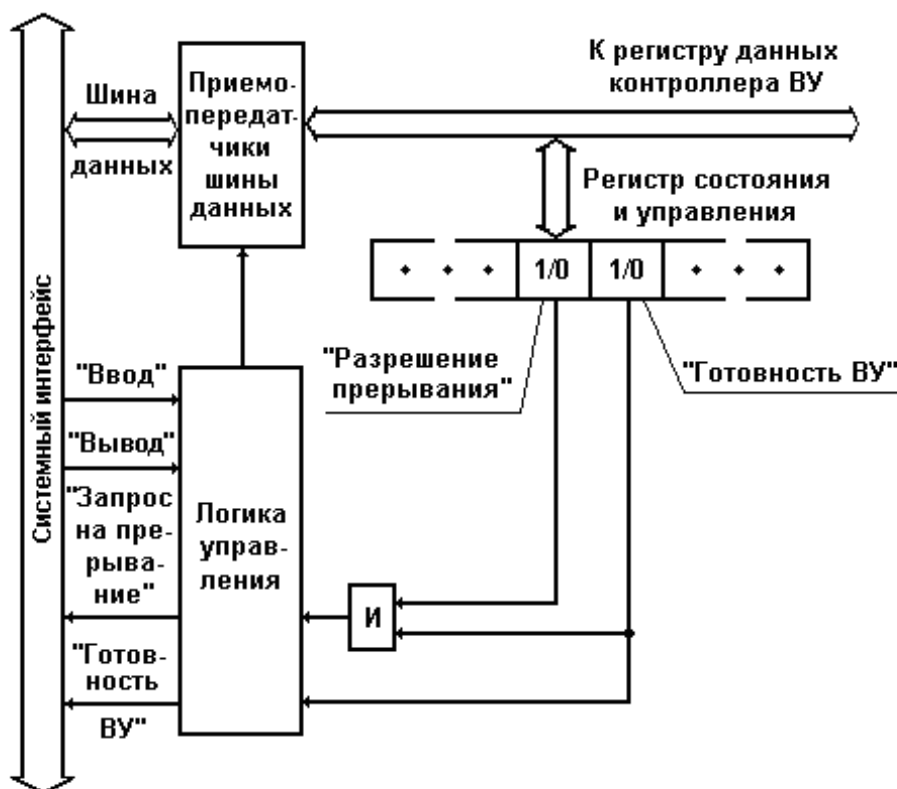


Рис. 38. Фрагмент блок-схемы контроллера ВУ с разрядом "Разрешение прерывания" в регистре состояния и управления

В микроЭВМ обычно используется одноуровневая система прерываний, т. е. сигналы "Запрос на прерывание" от всех ВУ поступают на один вход процессора. Поэтому возникает проблема идентификации ВУ, запросившего обслуживание, и реализации заданной очередности (приоритета) обслуживания ВУ при одновременном поступлении нескольких сигналов прерывания. Существуют два основных способа идентификации ВУ, запросивших обслуживания:

- программный опрос регистров состояния (разряд "Готовность ВУ") контроллеров всех ВУ;
- использование векторов прерывания.

Организация прерываний с программным опросом готовности предполагает наличие в памяти микроЭВМ единой подпрограммы обслуживания прерываний от всех внешних устройств. Структура такой подпрограммы приведена на рис. 39.

Обслуживание ВУ с помощью единой подпрограммы обработки прерываний производится следующим образом. В конце последнего машинного цикла выполнения очередной команды основной программы процессор проверяет наличие требования прерывания от ВУ. Если сигнал прерывания есть и в процессоре прерывание разрешено, то процессор переключается на выполнение подпрограммы обработки прерываний.

После сохранения содержимого регистров процессора, используемых в подпрограмме, начинается последовательный опрос регистров состояния контроллеров всех ВУ, работающих в режиме прерывания. Как только подпрограмма обнаружит готовое к обмену ВУ, сразу выполняются действия по его обслуживанию. Завершается подпрограмма обработки прерывания после опроса готовности всех ВУ и восстановления содержимого регистров процессора.

Приоритет ВУ в микроЭВМ с программным опросом готовности внешнего устройства однозначно определяется порядком их опроса в подпрограмме обработки прерываний. Чем раньше в подпрограмме опрашивается готовность ВУ, тем меньше время реакции на его запрос и выше приоритет. Необходимость проверки готовности всех внешних устройств существенно увеличивает время обслуживания тех ВУ, которые опрашиваются последними. Это является основным недостатком рассматриваемого способа организации прерываний. Поэтому обслуживание прерываний с опросом готовности ВУ используется только в тех случаях, когда отсутствуют жесткие требования на время обработки сигналов прерывания внешних устройств.

Организация системы прерываний в микроЭВМ с использованием векторов прерываний позволяет устранить указанный недостаток. При такой организации системы прерываний ВУ, запросившее обслуживания, само идентифицирует себя с помощью вектора прерывания – адреса ячейки основной памяти микроЭВМ, в которой хранится либо первая команда подпрограммы обслуживания прерывания данного ВУ, либо адрес начала такой подпрограммы.

Таким образом, процессор, получив вектор прерывания, сразу переключается на выполнение требуемой подпрограммы обработки прерывания. В микроЭВМ с векторной системой прерывания каждое ВУ должно иметь собственную подпрограмму обработки прерывания.

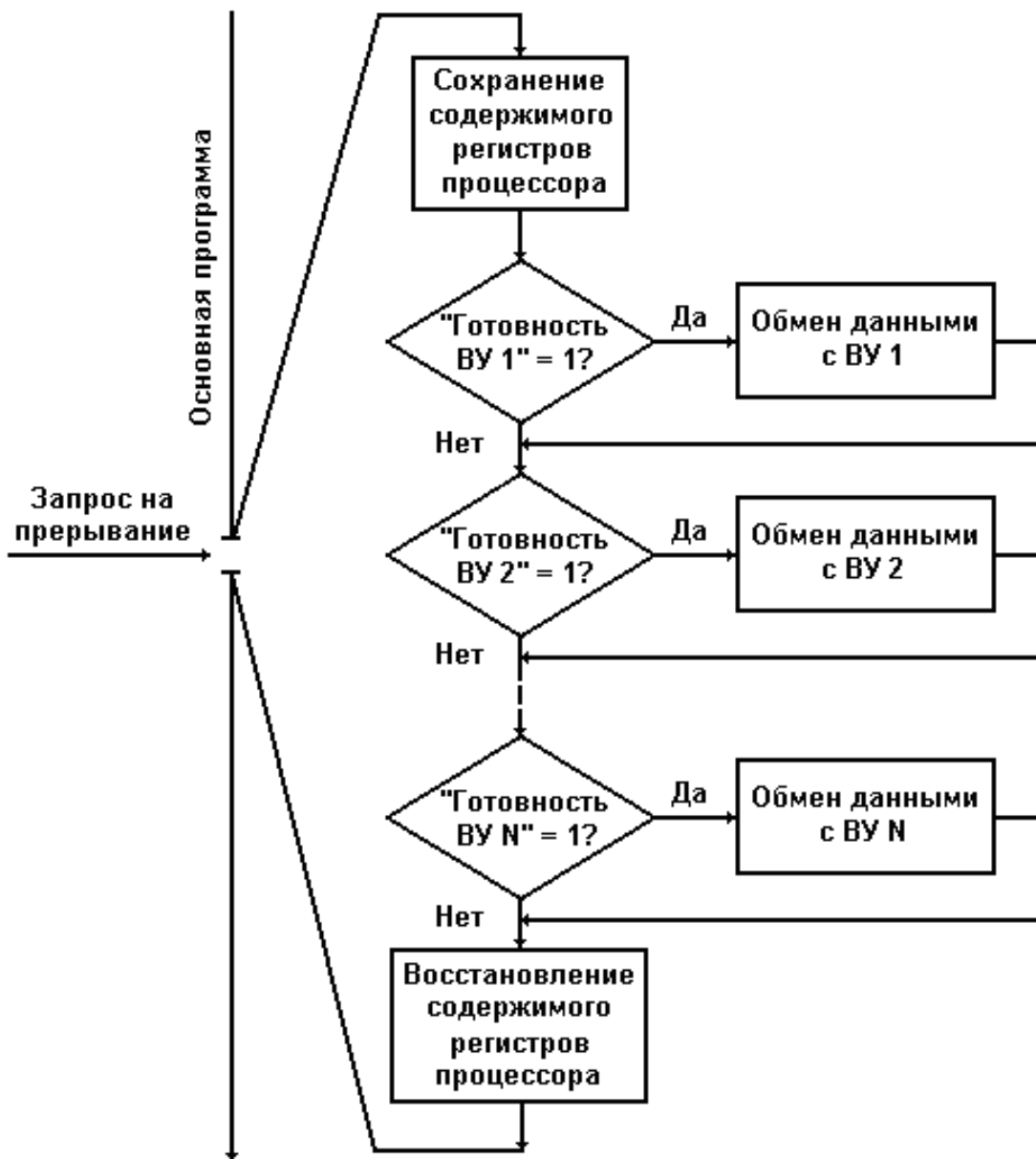


Рис. 39. Структура единой программы обработки прерываний и ее связь с основной программой

Различают векторные системы с интерфейсным и внеинтерфейсным вектором. В первом случае вектор прерывания формирует контроллер ВУ, запросившего обслуживания, во втором – контроллер прерываний, общий для всех устройств, работающих в режиме прерываний (IBM-совместимые персональные компьютеры).

Рассмотрим организацию векторной системы с интерфейсным вектором [5, 6]. Вектор прерывания выдается контроллером не одновременно с запросом на прерывание, а только по разрешению процессора, как это реализовано в схеме на рис. 40. Это делается для того, чтобы исключить одновременную выдачу векторов прерывания от нескольких ВУ. В ответ на сигнал контроллера ВУ "Запрос на прерывание" процессор формирует управляющий сигнал "Предоставление прерывания (вх.)", который разрешает контроллеру ВУ, запросившему обслуживание, выдачу вектора прерывания в шину адреса системного интерфейса. Для этого в контроллере используются регистр вектора прерывания и схема совпадения ИЗ. Регистр вектора прерывания обычно реализуется с помощью перемычек или переключателей, что позволяет пользователю устанавливать для конкретных ВУ требуемые значения векторов прерывания.

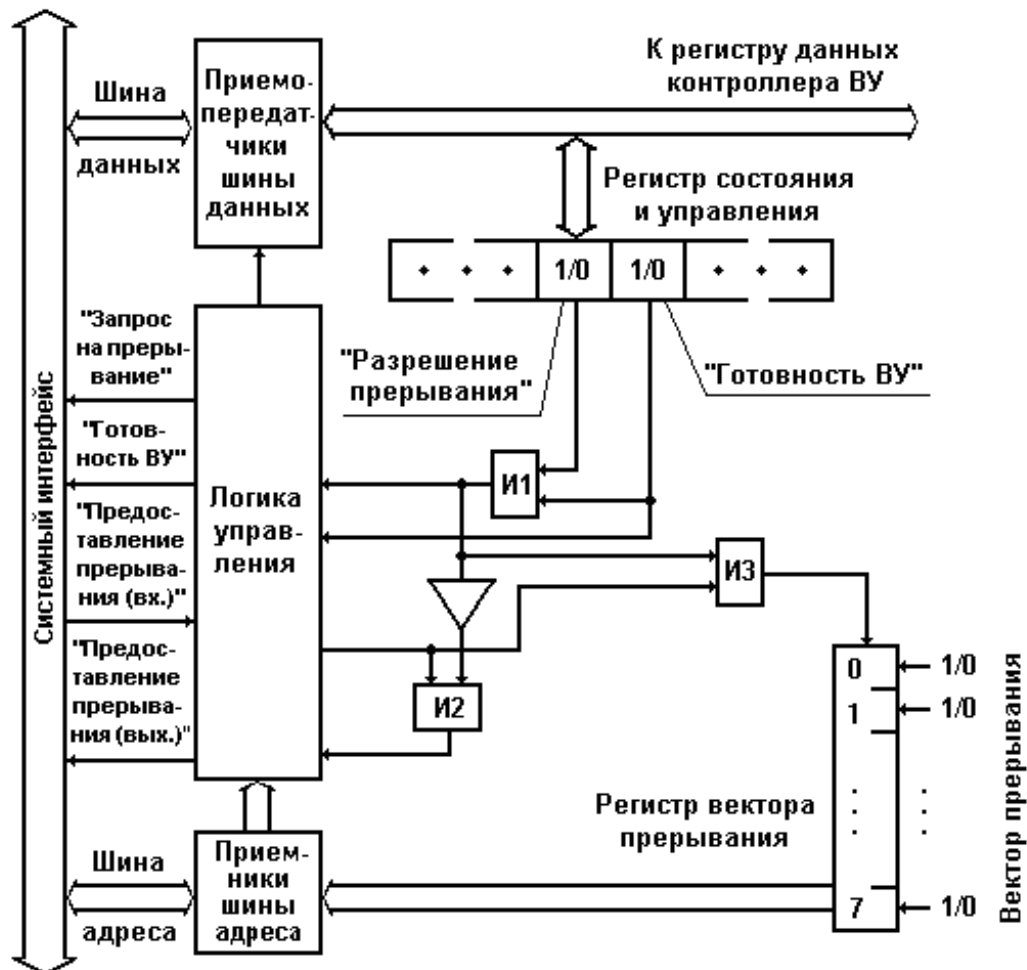


Рис. 40. Формирование векторов прерывания в контроллере ВУ

Управляющий сигнал "Предоставление прерывания (вых.)" формируется в контроллере ВУ с помощью схемы совпадения И2. Этот

сигнал используется для организации последовательного аппаратного опроса готовности ВУ и реализации тем самым требуемых приоритетов ВУ. Процессор при поступлении в него по общей линии системного интерфейса "Запрос на прерывание" сигнала прерывания формирует управляющий сигнал "Предоставление прерывания (вх.)", который поступает сначала в контроллер ВУ с наивысшим приоритетом (рис. 41). Если это устройство не требовало обслуживания, то его контроллер пропускает сигнал "Предоставление прерывания" на следующий контроллер, иначе дальнейшее распространение сигнала прекращается и контроллер выдает вектор прерывания на адресноинформационную шину.

Аппаратный опрос готовности ВУ производится гораздо быстрее, нежели программный. Но если обслуживания запросили одновременно два или более ВУ, обслуживание менее приоритетных ВУ будет отложено на время обслуживания более приоритетных, как и в системе прерывания с программным опросом.



Рис. 41. Реализация приоритетов ВУ в микроЭВМ с векторной системой прерываний. С интерфейсным вектором (ППР (вх.) – "Предоставление прерывания (входной)"; "ППР (вых.) – Предоставление прерывания (выходной)")

Рассмотренная векторная система прерываний практически полностью соответствует системе прерываний, реализованной в микроЭВМ "Электроника-60". Восемьразрядный вектор прерывания в "Электронике-60" указывает одну из ячеек памяти с адресами от 0 до 255, в которой размещается адрес начала подпрограммы обработки прерывания. В следующей за указанной вектором прерывания ячейке памяти хранится новое содержимое регистра состояния процессора, загружаемое в него при переключении на подпрограмму обработки прерывания. Один из бит нового содержимого регистра состояния процессора запрещает или разрешает прерывания от других ВУ, что позволяет ВУ с

более высоким приоритетом прерывать подпрограммы обслуживания ВУ с меньшим приоритетом и наоборот.

Векторная система с внеинтерфейсным вектором прерывания используется в IBM-совместимых персональных компьютерах. В этих компьютерах контроллеры внешних устройств не имеют регистров для хранения векторов прерывания, а для идентификации устройств, запросивших обслуживания, используется общий для всех ВУ контроллер прерываний. Ниже приведен пример контроллера прерываний INTEL 8259A.

БИС программируемого контроллера прерываний (ПКП) представляет собой устройство, реализующее до восьми уровней запросов на прерывания с возможностью программного маскирования и изменения порядка обслуживания прерываний. За счет каскадного включения БИС ПКП число уровней прерывания может быть расширено до 64 (в архитектуре персонального компьютера IBM PC AT).

Структурная схема ПКП приведена на рис. 42 [2, 5, 6].

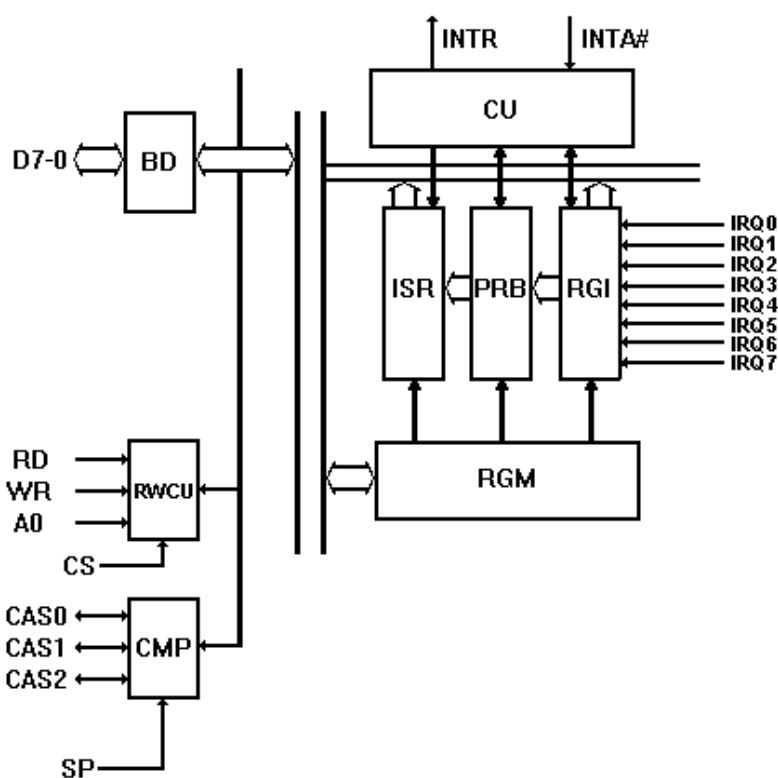


Рис. 42. Контроллер прерываний Intel 8259A

В состав БИС входят:

RGI – регистр запретов прерываний; хранит все уровни, на которые поступают запросы IRQx;

PRB – схема принятия решений по приоритетам; схема идентифицирует приоритет запросов и выбирает запрос с наивысшим приоритетом;

ISR – регистр обслуживаемых прерываний; сохраняет уровни запросов прерываний, находящиеся на обслуживании ПКП;

RGM – регистр маскирования прерываний; обеспечивает запрещение одной или нескольких линий запросов прерывания;

BD – буфер данных; предназначен для сопряжения ПКП с системной шиной данных;

RWCU – блок управления записью/чтением; принимает управляющие сигналы от микропроцессора и задает режим функционирования ПКП;

СМР – схема каскадного буфера-компаратора; используется для включения в систему нескольких ПКП;

CU – схема управления; вырабатывает сигналы прерывания и формирует трехбайтовую команду CALL для выдачи на шину данных.

Установка ПКП в исходное состояние и "настройка" его на определенный режим обслуживания прерываний происходит с помощью двух типов команд: команд инициализации (ICW) и команд управления операциями (OCW).

Программируемый контроллер прерываний (ПКП) имеет 16 входов запросов прерываний (IRQ 0 – IRQ 15). Контроллер состоит из двух каскадно включенных контроллеров – выход INTR (запрос на прерывание) второго контроллера подключен ко входу IRQ 2 первого контроллера. В качестве примера отметим, что к линии IRQ 0 подключен системный таймер, к линии IRQ 1 – клавиатура, к линии IRQ 8 – часы реального времени и т. д.

Упрощенная схема взаимодействия контроллера прерываний с процессором и контроллером шины имеет следующий вид (см. рис. 43).

Эта схема функционирует следующим образом. Пусть в некоторый момент времени контроллер клавиатуры с помощью единичного сигнала по линии IRQ 1 известил контроллер прерываний о своей готовности к обмену. В ответ на запрос контроллер прерываний генерирует сигнал INTR (запрос на прерывание) и посылает его на соответствующий вход процессора. Процессор, если маскируемые прерывания разрешены (т. е. установлен флаг прерываний IF в регистре флагов процессора), посылает на контроллер шины сигналы R# – чтение, C# – управление и IO# – ввод/вывод, определяющие тип цикла шины. Контроллер шины, в свою очередь, генерирует два сигнала подтверждения прерывания INTA# и направляет их на контроллер прерываний. По

второму импульсу контроллер прерываний выставляет на шину данных восьмибитный номер вектора прерывания, соответствующий данной линии IRQ.

В режиме реального адреса ("реальном" режиме) векторы прерываний хранятся в таблице векторов прерываний, которая находится в первом килобайте оперативной памяти. Под каждый вектор отведено 4 байта (2 байта под адрес сегмента и 2 байта под смещение), т. е. в таблице может содержаться 256 векторов. Адрес вектора в таблице – номер вектора * 4.

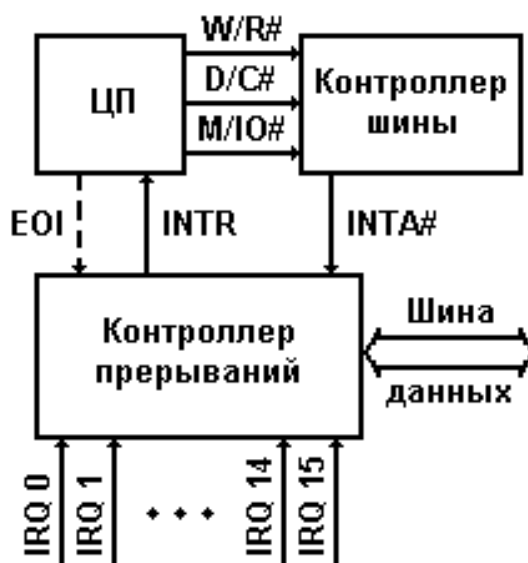


Рис. 43. Упрощенная схема взаимодействия контроллера прерываний с процессором и контроллером шины в IBM-совместимых ПК

Далее процессор считывает номер вектора прерывания. Сохраняет в стеке содержимое регистра флагов, сбрасывает флаг прерываний IF и помещает в стек адрес возврата в прерванную программу (регистры CS и IP). После этого процессор извлекает из таблицы векторов прерываний адрес подпрограммы обработки прерываний для данного устройства и приступает к ее выполнению.

Процедура обработки аппаратного прерывания должна завершаться командой конца прерывания EOI (End of Interruption), посылаемой контроллеру прерываний. Для этого необходимо записать байт 20h в порт 20h (для первого контроллера) и в порт A0h (для второго).

В IBM PC/XT/AT используется режим прерываний с фиксированными приоритетами. Высшим приоритетом обладает запрос по линии

IRQ 0, низшим – IRQ 7. Так как второй контроллер подключен к линии IRQ 2 первого контроллера, то приоритеты линий IRQ в порядке убывания приоритета располагаются следующим образом: IRQ 0, IRQ 1, IRQ 8 – IRQ 15, IRQ 3 – IRQ 7. Если запрос на обслуживание посылают одновременно два устройства с разными приоритетами, то контроллер обслуживает запрос с большим приоритетом, а запрос с меньшим приоритетом блокирует. Блокировка сохраняется до получения команды EOI.

5.5.3. Организация прямого доступа к памяти

Одним из способов обмена данными с ВУ является обмен в режиме прямого доступа к памяти (ПДП, DMA – direct memory access). В этом режиме обмен данными между ВУ и основной памятью микроЭВМ происходит без участия процессора. Обменом в режиме ПДП управляет не программа, выполняемая процессором, а электронные схемы, внешние по отношению к процессору. Обычно схемы, управляющие обменом в режиме ПДП, размещаются в специальном контроллере, который называется контроллером прямого доступа к памяти [5, 6].

Обмен данными в режиме ПДП позволяет использовать в микроЭВМ быстродействующие внешние запоминающие устройства, такие, например, как накопители на жестких магнитных дисках, поскольку ПДП может обеспечить время обмена одним байтом данных между памятью и ВЗУ, равное циклу обращения к памяти.

Для реализации режима прямого доступа к памяти необходимо обеспечить непосредственную связь контроллера ПДП и памяти микроЭВМ. Для этой цели можно было бы использовать специально выделенные шины адреса и данных, связывающие контроллер ПДП с основной памятью. Но такое решение нельзя признать оптимальным, так как это приведет к значительному усложнению микроЭВМ в целом, особенно при подключении нескольких ВЗУ. В целях сокращения количества линий в шинах микроЭВМ контроллер ПДП подключается к памяти посредством шин адреса и данных системного интерфейса. При этом возникает проблема совместного использования шин системного интерфейса процессором и контроллером ПДП. Можно выделить два основных способа ее решения: реализация обмена в режиме ПДП с "захватом цикла" и в режиме ПДП с блокировкой процессора.

Существуют две разновидности прямого доступа к памяти с "захватом цикла". Наиболее простой способ организации ПДП состоит в том, что для обмена используются те машинные циклы процессора, в

которых он не обменивается данными с памятью. В такие циклы контроллер ПДП может обмениваться данными с памятью, не мешая работе процессора. Однако возникает необходимость выделения таких циклов, чтобы не произошло временного перекрытия обмена ПДП с операциями обмена, инициируемыми процессором. В некоторых процессорах формируется специальный управляющий сигнал, указывающий циклы, в которых процессор не обращается к системному интерфейсу. При использовании других процессоров для выделения таких циклов необходимо применение в контроллерах ПДП специальных селективных схем, что усложняет их конструкцию. Применение рассмотренного способа организации ПДП не снижает производительности микроЭВМ, но при этом обмен в режиме ПДП возможен только в случайные моменты времени одиночными байтами или словами.

Более распространенным является ПДП с "захватом цикла" и принудительным отключением процессора от шин системного интерфейса. Для реализации такого режима ПДП системный интерфейс микроЭВМ дополняется двумя линиями для передачи управляющих сигналов "Требование прямого доступа к памяти" (ТПДП, DRQ – Dma ReQuest) и "Предоставление прямого доступа к памяти" (ППДП, DACK – Dma ACKnowledged).

Управляющий сигнал ТПДП формируется контроллером прямого доступа к памяти. Процессор, получив этот сигнал, приостанавливает выполнение очередной команды, не дожидаясь ее завершения, выдает на системный интерфейс управляющий сигнал ППДП и отключается от шин системного интерфейса. С этого момента все шины системного интерфейса управляются контроллером ПДП. Контроллер ПДП, используя шины системного интерфейса, осуществляет обмен одним байтом или словом данных с памятью микроЭВМ и затем, сняв сигнал ТПДП, возвращает управление системным интерфейсом процессору. Как только контроллер ПДП будет готов к обмену следующим байтом, он вновь "захватывает" цикл процессора и т. д. В промежутках между сигналами ТПДП процессор продолжает выполнять команды программы. Тем самым выполнение программы замедляется, но в меньшей степени, чем при обмене в режиме прерываний.

Применение в микроЭВМ обмена данными с ВУ в режиме ПДП всегда требует предварительной подготовки, а именно: для каждого ВУ необходимо выделить область памяти, используемую при обмене, и указать ее размер, т. е. количество записываемых в память или читаемых из памяти байт (слов) информации. Следовательно, контроллер ПДП должен обязательно иметь в своем составе регистр адреса и счетчик байт (слов). Перед началом обмена с ВУ в режиме ПДП процессор

должен выполнить программу загрузки. Эта программа обеспечивает запись в указанные регистры контроллера ПДП начального адреса выделенной ВУ памяти и ее размера в байтах или словах в зависимости от того, какими порциями информации ведется обмен. Сказанное не относится к начальной загрузке программ в память в режиме ПДП. В этом случае содержимое регистра адреса и счетчика байт слов устанавливается переключателями или перемычками непосредственно на плате контроллера.

Блок-схема простого контроллера ПДП, обеспечивающего ввод данных в память микроЭВМ по инициативе ВУ в режиме ПДП "Захват цикла", приведена на рис. 44.

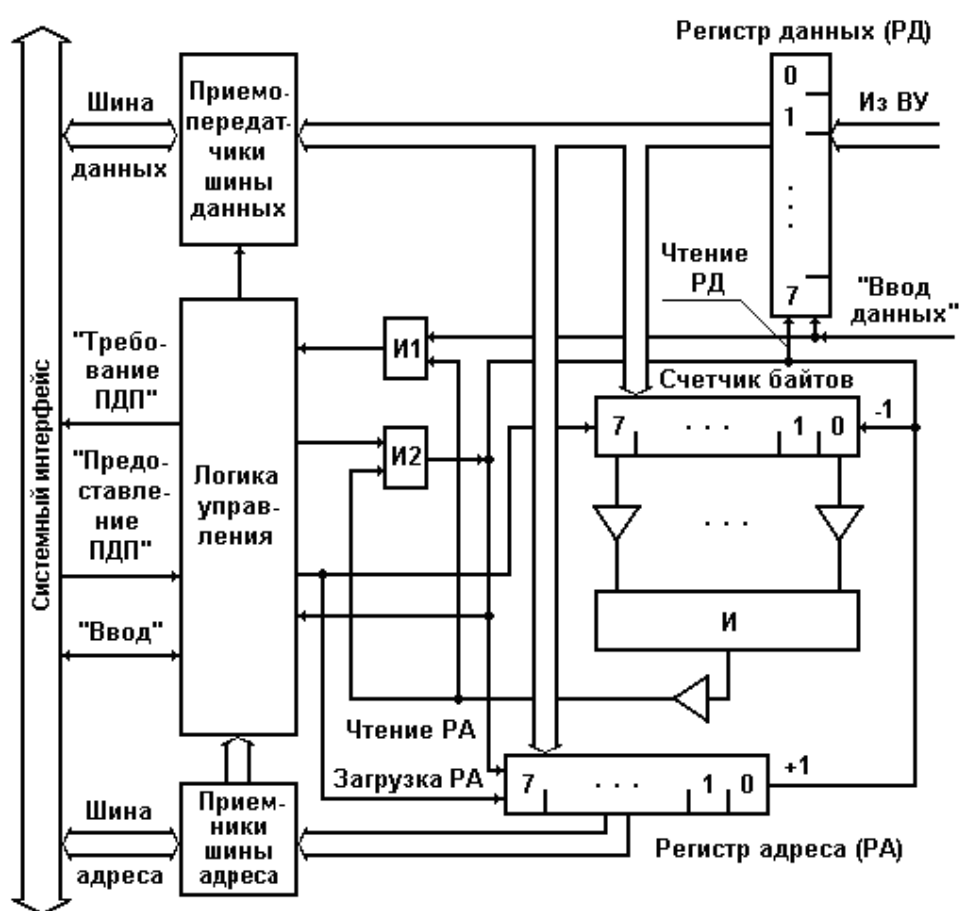


Рис. 44. Контроллер ПДП для ввода данных из ВУ в режиме "Захват цикла" и отключением процессора от шин системного интерфейса

Перед началом очередного сеанса ввода данных из ВУ процессор загружает в регистры его контроллера следующую информацию: в счетчик байт – количество принимаемых байт данных, а в регистр адреса – начальный адрес области памяти для вводимых данных. Тем са-

мым контроллер подготавливается к выполнению операции ввода данных из ВУ в память микроЭВМ в режиме ПДП.

Байты данных из ВУ поступают в регистр данных контроллера в постоянном темпе. При этом каждый байт сопровождается управляющим сигналом из ВУ "Ввод данных", который обеспечивает запись байта данных в регистр данных контроллера. По этому же сигналу и при ненулевом состоянии счетчика байт контроллер формирует сигнал ПДП. По ответному сигналу процессора ПДП контроллер выставляет на шины адреса и данных системного интерфейса содержимое своих регистров адреса и данных, соответственно. Формируя управляющий сигнал "Вывод", контроллер ПДП обеспечивает запись байта данных из своего регистра данных в память микроЭВМ. Сигнал ПДП используется в контроллере и для модификации счетчика байт и регистра адреса. По каждому сигналу ПДП из содержимого счетчика байт вычитается единица, и как только содержимое счетчика станет равно нулю, контроллер прекратит формирование сигналов "Требование прямого доступа к памяти".

На примере простого контроллера ПДП мы рассмотрели только процесс подготовки контроллера и непосредственно передачу данных в режиме ПДП. На практике любой сеанс обмена данными с ВУ в режиме ПДП всегда иницируется программой, выполняемой процессором, и включает два следующих этапа.

1. На этапе подготовки ВУ к очередному сеансу обмена процессор в режиме программно-управляемого обмена опрашивает состояние ВУ (проверяет его готовность к обмену) и посылает в ВУ команды, обеспечивающие подготовку ВУ к обмену. Такая подготовка может сводиться, например, к перемещению головок на требуемую дорожку в накопителе на жестком диске. Затем выполняется загрузка регистров контроллера ПДП. На этом подготовка к обмену в режиме ПДП завершается и процессор переключается на выполнение другой программы.

2. Обмен данными в режиме ПДП начинается после завершения подготовительных операций в ВУ по инициативе либо ВУ, как это было рассмотрено выше, либо процессора. В этом случае контроллер ПДП необходимо дополнить регистром состояния и управления, содержимое которого будет определять режим работы контроллера ПДП. Один из разрядов этого регистра будет иницировать обмен данными с ВУ. Загрузка информации в регистр состояния и управления контроллера ПДП производится программным путем.

Наиболее распространенным является обмен в режиме прямого доступа к памяти с блокировкой процессора. Он отличается от ПДП с "захватом цикла" тем, что управление системным интерфейсом переда-

ется контроллеру ПДП не на время обмена одним байтом, а на время обмена блоком данных. Такой режим ПДП используется в тех случаях, когда время обмена одним байтом с ВУ сопоставимо с циклом системной шины.

В микроЭВМ можно использовать несколько ВУ, работающих в режиме ПДП. Предоставление таким ВУ шин системного интерфейса для обмена данными производится на приоритетной основе. Приоритеты ВУ реализуются так же, как и при обмене данными в режиме прерывания, но вместо управляющих сигналов "Требование прерывания" и "Предоставление прерывания" (рис. 40) используются сигналы "Требование прямого доступа" и "Предоставление прямого доступа", соответственно.

Библиографический список

1. Балашов Е. П., Григорьев В. Л., Петров Г. А. Микро- и мини-ЭВМ. – Л.: Энергоатомиздат, 1984. – 376 с.
2. Ершова Н. Ю., Иващенко О. Н., Курсков С. Ю. Микропроцессоры. – Санкт-Петербург, 2002.
3. Микропроцессоры: в 3-х кн. / под ред. С. В. Преснухина. – М.: Высшая школа, 1986. – Кн.1. – 495 с. – Кн. 2. – 383 с. – Кн. 3. – 351 с.
4. Гивоне Д., Россер Р. Микропроцессоры и микрокомпьютеры: Вводный курс / пер. с англ. – М.: Мир, 1983. – 463 с.
5. Морисита И. Аппаратные средства микроЭВМ / пер. с япон. – М.: Мир, 1988. – 279 с.
6. Гибсон Г., Лю Ю. Ч. Аппаратные и программные средства микроЭВМ / пер. с англ. В. Л. Григорьева; под ред. В. В. Сташина. – М.: Финансы и статистика, 1983. – 255 с.
7. Напрасник М. В. Микропроцессоры и микроЭВМ: Учебное пособие для техникумов. – М.: Высш. шк., 1989. – 192 с.
8. Токхайм Р. Микропроцессоры: Курс и упражнения / под ред. А. Г. Грасевича. – М.: Энергоатомиздат, 1987. – 338 с.
9. Лебедев О. Н. Применение микросхем памяти в электронных устройствах: справ. пособие. – М.: Радио и связь, 1994. – 216 с.

Список рекомендуемой литературы

1. Басманов А. С., Широков Ю. Ф. Микропроцессоры и однокристалльные микроЭВМ: Номенклатура и функциональные возможности / под ред. В. Г. Домрачева. – М.: Энергоатомиздат, 1988. – 127 с.
2. 1. Вуд А. Микропроцессоры в вопросах и ответах / пер. с англ. – М.: Энергоатомиздат, 1985. – 185 с.
3. Интерфейсы систем обработки данных: справочник / под ред. А. А. Мячева. – М.: Радио и связь, 1989.
4. Комаров А. В. Введение в микропроцессоры: Конспект лекций по курсу «Микропроцессорные устройства». – Обнинск: ИАТЭ, 1998.
5. Мячев А. А., Иванов В. В. Интерфейсы вычислительных систем на базе мини- и микроЭВМ / под ред. Б. Н. Наумова. – М.: Радио и связь, 1986. – 248 с.
6. Нестеров П. В. Микропроцессоры. Архитектура и ее оценка. – М.: Высшая школа, 1984. – 104 с.
7. Уокерли Дж. Архитектура и программирование микроЭВМ: в 2-х кн. / пер. с англ. – М.: Мир, 1984. – Кн. 1. – 486 с. – Кн. 2. – 359 с.
8. Хосе М. Ангуло. Микропроцессоры: Архитектура, программирование и проектирование систем. – Тбилиси: Ганатлеба, 1989.

Сергей Николаевич Ливенцов
Александр Даниилович Вильнин
Горюнов Алексей Германович

ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

Научный редактор
кандидат технических наук, доцент В. Ф. Дядик


Редактор М.В. Пересторонина

Подписано к печати 21.03.2007. Формат 60x84/16. Бумага «Классика».
Печать RISO. Усл.печ.л. 6,86. Уч.-изд.л. 6,21.
Заказ . Тираж 100 экз.



Томский политехнический университет
Система менеджмента качества
Томского политехнического университета
сертифицирована
NATIONAL QUALITY ASSURANCE по стандарту
ISO 9001:2000



ИЗДАТЕЛЬСТВО  ТПУ. 634050, г. Томск, пр. Ленина, 30.