

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

В. П. ГАЛАС

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И ТЕЛЕКОММУНИКАЦИИ

Учебник

В двух частях

Часть 1. Вычислительные системы



Владимир 2016

УДК 658.5
ББК 32.9
Г15

Рецензенты:

Кандидат технических наук, доцент
доцент кафедры вычислительной техники
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
В. Б. Буланкин

Кандидат технических наук
технический директор ООО «НПП "Энергоприбор"»
С. А. Кокорин

Печатается по решению редакционно-издательского совета ВлГУ

Галас, В. П.

Г15 Вычислительные системы, сети и телекоммуникации : учебник.
В 2 ч. Ч. 1. Вычислительные системы / В. П. Галас ; Владим. гос. ун-т
им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2016. – 232 с. –
ISBN 978-5-9984-0730-7 (ч. 1). – ISBN 978-5-9984-0731-4.

Излагаются история возникновения, принципы построения, архитектура, функциональная и структурная организации ЭВМ. Рассматриваются функционирование, программное обеспечение и перспективы развития основных устройств и элементов вычислительных машин и систем. Основная цель учебника – познакомить студентов и специалистов, профессиональная деятельность которых связана с применением вычислительных машин и систем, с широким кругом вопросов, касающихся принципов построения и организации функционирования компьютеров, вычислительных систем, технологии ввода-вывода и обработки информации. Издание содержит интерактивное вложение на компакт-диске, а именно электронный учебник, рекомендованный УМО вузов РФ по образованию в области радиотехники, электроники, биомедицинской техники и автоматизации в качестве электронного учебника для студентов высших учебных заведений, обучающихся по направлению подготовки бакалавров 27.03.04 (220400) «Управление в технических системах». Диск включает теоретический материал, разделенный на отдельные главы, с системой гиперссылок, видеоматериалом, контрольными вопросами и встроенными программами тестирования знаний. Содержит блок контрольных мероприятий, включающий программы для реализации рейтинг-контроля в процессе обучения, инструкцию по использованию и материалы справочного характера.

Предназначен для студентов и слушателей всех форм обучения с использованием мультимедийных и дистанционных образовательных технологий, а также для преподавателей высших и средних специальных учебных заведений.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Табл. 21. Ил. 106. Библиогр.: 20 назв.

УДК 658.5
ББК 32.9

ISBN 978-5-9984-0730-7 (ч. 1)
ISBN 978-5-9984-0731-4

© ВлГУ, 2016

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	7
ВВЕДЕНИЕ	9
Глава 1. ИСТОРИЯ РАЗВИТИЯ, ТИПЫ И ПОКОЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И СИСТЕМ	11
1.1. Понятие ЭВМ, основные типы вычислительных машин.....	11
1.2. Нулевое поколение (механические вычислители)	12
1.3. Первое поколение (вычислители на электронных лампах)	15
1.4. Второе поколение (компьютеры на транзисторах)	18
1.5. Третье поколение (компьютеры на интегральных схемах)	20
1.6. Четвертое поколение (компьютеры на СБИС)	23
1.7. Пятое поколение (компьютеры с искусственным интеллектом)	29
Вопросы к компьютерному тестированию	31
Глава 2. АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ	34
2.1. Системы счисления.....	34
2.2. Перевод целых чисел.....	36
2.3. Перевод дробных чисел.....	37
2.4. Представление информации в ЭВМ	38
2.4.1. Представление числовой информации	38
2.4.2. Представление других видов информации	40
2.5. Кодирование информации.....	41
2.6. Выполнение арифметических операций.....	44
2.6.1. Машинные коды.....	45
2.6.2. Арифметические операции над числами с фиксированной точкой.....	47
2.6.3. Арифметические операции над двоичными числами с плавающей точкой.....	50
2.6.4. Арифметические операции над двоично-десятичными кодами чисел.....	53
Вопросы к компьютерному тестированию	54

Глава 3. ЛОГИЧЕСКИЕ ОСНОВЫ ЭВМ.....	56
3.1. Основные понятия алгебры логики.....	56
3.2. Связь алгебры логики с системой кодирования данных ЭВМ.....	59
3.3. Законы алгебры логики	63
3.4. Понятие о минимизации логических функций.....	64
3.5. Техническая интерпретация логических функций.....	66
Вопросы к компьютерному тестированию	68
Глава 4. ЭЛЕМЕНТНАЯ БАЗА ЭВМ.....	70
4.1. Классификация элементов и узлов ЭВМ.....	70
4.2. Комбинационные схемы.....	72
4.3. Последовательностные схемы	81
4.4. Развитие элементной базы ЭВМ	89
Вопросы к компьютерному тестированию	95
Глава 5. ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИИ ЭВМ	97
5.1. Общие принципы функциональной и структурной организаций ЭВМ.....	97
5.2. Совместимость ЭВМ	101
5.3. Функциональные компоненты ЭВМ и их взаимосвязь	102
5.4. Центральные устройства ЭВМ	106
5.4.1. Системная (материнская) плата.....	106
5.4.2. Микропроцессорный набор	112
5.4.3. Узлы, выполняющие служебные функции.....	118
Вопросы к компьютерному тестированию	120
Глава 6. ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР ЭВМ	122
6.1. Основные характеристики процессора.....	122
6.2. Микропроцессоры типа <i>CISC</i>	128
6.3. Микропроцессоры типа <i>RISC</i>	129
6.4. Микропроцессоры типа <i>VLIW</i> и <i>MISC</i>	130

6.5. Физическая и функциональная структура микропроцессора	130
6.6. Устройство управления	131
6.7. Арифметико-логическое устройство	132
6.8. Микропроцессорная память.....	134
6.9. Интерфейсная часть ЦП	135
6.10. Порты ввода-вывода	135
6.11. Функциональные характеристики ЦП.....	136
6.12. Система команд ЦП	137
6.13. Взаимодействие элементов при работе ЦП	138
6.14. Работа центрального процессора при выполнении задания пользователя	140
6.15. Работа микропроцессора при выполнении программного прерывания	143
Вопросы к компьютерному тестированию	146

Глава 7. АДАПТЕРЫ. ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЭВМ.....

7.1. Адаптеры.....	148
7.2. Запоминающие устройства ЭВМ	150
7.2.1. Состав, устройство и принцип действия основных запоминающих устройств	150
7.2.2. Внешние запоминающие устройства	163
Вопросы к компьютерному тестированию	172

Глава 8. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

8.1. Структура программного обеспечения ЭВМ.....	174
8.2. Общее программное обеспечение	175
8.3. Операционные системы.....	176
8.4. Системы автоматизации программирования	180
8.5. Комплекс программ технического обслуживания	185
8.6. Специальное программное обеспечение. Пакеты программ.....	187
Вопросы к компьютерному тестированию	191

Глава 9. ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ.....	193
9.1. Общие положения и классификация вычислительных систем.....	193
9.2. Архитектура вычислительных систем.....	197
9.3. Кластеризация вычислительных систем.....	200
9.4. Организация функционирования вычислительных систем.....	203
9.5. Методы комплексирования вычислительных систем.....	204
9.6. Суперкомпьютеры и особенности их архитектуры	209
Вопросы к компьютерному тестированию	213
 ЗАКЛЮЧЕНИЕ	215
 СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....	217
 ГЛОССАРИЙ	219

ПРЕДИСЛОВИЕ

Цель написания настоящего учебника – систематическое изложение вопросов, связанных с хронологией создания, классификацией, структурной и функциональной организацией, технической реализацией, программным обеспечением и характеристиками вычислительных машин, систем, сетей и телекоммуникаций, в тех объемах, которые отведены на дисциплину «Вычислительные системы, сети и телекоммуникации» в учебном плане базового высшего образования по направлениям «Управление в технических системах» и «Прикладная информатика» для бакалавров.

Задача дисциплины – изучение студентами основ построения и организации функционирования вычислительных машин, систем и коммуникационных сетей, формирование знаний в области структурной организации, аппаратных и программных ресурсов, освоение способов эффективного применения современных технических средств для решения информационных задач, обучение правильному выбору и эксплуатации ЭВМ, вычислительных сетей и телекоммуникаций в соответствии с требованиями, предъявляемыми к информационным системам.

Материалы учебника базируются на публикациях отечественных и зарубежных авторов и материале курса лекций, читаемых автором студентам Владимирского государственного университета в течение ряда лет. Кроме теоретической части приводятся вопросы для тестирования объема изученного с целью контроля приобретенных знаний в области вычислительной техники, организации и эксплуатации вычислительных машин и сетей.

Учебник представляет собой издание в двух частях, первая из которых включает материал по вычислительным машинам и системам,

вторая – по вычислительным сетям и телекоммуникациям. Обе части комплектуются интерактивным вложением на компакт-диске. Диск, в свою очередь, предлагает для изучения теоретический материал, разделенный на отдельные главы с вопросами для самопроверки, содержит систему меню и гиперссылок, электронную книгу с эффектом перелистывания страниц и тематическим поиском, глоссарий, списки библиографической и рекомендуемой литературы, дополнительный справочный и видеоматериал, встроенные программы тестирования знаний, блок контрольных мероприятий с программами для реализации тестирования в процессе обучения, а также все необходимые инструкции для работы с диском.

Автор надеется, что учебник будет полезен студентам и слушателям всех форм обучения, включая использующих мультимедийные и дистанционные образовательные технологии, а также преподавателям высших и средних специальных учебных заведений, и с благодарностью примет все замечания, пожелания и предложения по содержанию книги, которые следует направлять по электронному адресу издательства ВлГУ: *rio.vlgu@yandex.ru* или автора: *vpgalas@mail.ru*.

ВВЕДЕНИЕ

Вычислительные системы, сети и телекоммуникации в современном обществе самые востребованные ресурсы. Войдя в человеческую жизнь, они стали неотъемлемой частью нашей цивилизации. За исторически очень короткий срок компьютеры благодаря огромным успехам электроники проделали такой путь в своем техническом совершенствовании, масштабах применения и влияния на человеческое общество, с каким не сравнится любое другое изобретение человечества, включая атомную энергию и космическую технику. Да и последние не могли бы получить столь мощного развития без использования достижений вычислительной техники.

С развитием вычислительной техники расширяется сфера ее применения, изменяется и терминология. Термины «вычислительная машина», «вычислительная система», «вычислительная сеть» выросли из своего дословного толкования в части прилагательного «вычислительная». Уже давно названные объекты выполняют не только и не столько вычисления, сколько преобразования информации, а именно её накопление, хранение, организацию, толкование, то есть представляют собой фактически информационные системы.

Основная цель первой части учебника – познакомить студентов и специалистов, профессиональная деятельность которых связана с применением вычислительных машин и систем, с широким кругом вопросов, касающихся их истории возникновения, принципов построения, структурной и функциональной организации, технической реализации, технологиями ввода, вывода и обработки информации. Рассматриваются функционирование, программное обеспечение и перспективы развития основных устройств и элементов вычислительных машин и систем.

В начальных главах учебника представлены хронология создания и характеристики вычислительных машин и систем, даются краткие сведения об арифметических и логических основах, а также элементной базе ЭВМ.

В последующих главах приведены классификация вычислительных машин по различным признакам, структурная организация, основные классы компьютеров, рассмотрены принципы работы, конструктивное исполнение и важнейшие параметры основных элементов устройств вычислительной техники, программное обеспечение, технические средства обработки информации, адаптеры, запоминающие устройства и периферийное оборудование.

В заключительной главе изучаются компоненты информационно-вычислительных систем, обсуждаются вопросы эффективности и качества вычислительных систем. Существенное внимание уделено рассмотрению новых технических решений в сферах разработки и внедрения высокопроизводительных многопроцессорных и многомашинных вычислительных комплексов.

Освоение представленного учебного материала позволит получить базовые фундаментальные знания в динамично развивающихся областях вычислительной техники и информационно-коммуникационных технологий.

Изложенные в учебнике сведения могут найти применение при изучении ряда смежных дисциплин направлений 27.03.04 «Управление в технических системах» и 09.03.03 «Прикладная информатика», а также соответствующих разделов в дисциплинах «Информатика», «Информационные технологии», «Информационные системы» других направлений и специальностей.

Глава 1

ИСТОРИЯ РАЗВИТИЯ, ТИПЫ И ПОКОЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И СИСТЕМ



Рассматриваемые вопросы:

- 1.1. Понятие ЭВМ, основные типы вычислительных машин.
- 1.2. Нулевое поколение (механические вычислители).
- 1.3. Первое поколение (вычислители на электронных лампах).
- 1.4. Второе поколение (компьютеры на транзисторах).
- 1.5. Третье поколение (компьютеры на интегральных схемах).
- 1.6. Четвертое поколение (компьютеры на СБИС).
- 1.7. Пятое поколение (компьютеры с искусственным интеллектом).

1.1. Понятие ЭВМ, основные типы вычислительных машин

Электронная вычислительная машина – комплекс технических и программных средств, предназначенный для автоматизации подготовки и решения задач пользователей. Под пользователем понимают человека, в интересах которого проводится обработка данных на ЭВМ, – это заказчики вычислительных работ, программисты, операторы.

В настоящее время в мире произведены, работают и продолжают выпускаться миллионы вычислительных машин, относящихся к различным поколениям, типам, классам, отличающимся своими областями применения, техническими характеристиками и вычислительными возможностями.

Традиционно электронную вычислительную технику подразделяют на *аналоговую и цифровую*.

В *аналоговых вычислительных машинах (АВМ)* обрабатываемая информация представляется соответствующими значениями аналоговых величин: тока, напряжения, угла поворота какого-либо механизма и т.п.

Эти машины обеспечивают приемлемое быстродействие, но не высокую точность вычислений (0,001 – 0,01). Распространение подобных машин ограничено. Они используются в основном в проектных и научно-исследовательских учреждениях в составе различных стендов по отработке сложных образцов техники. По своему назначению их можно рассматривать как специализированные вычислительные машины.

В *цифровых вычислительных машинах (ЦВМ)* информация подвергается двоичному кодированию. Именно эти машины благодаря универсальным возможностям являются наиболее массовой вычислительной техникой.

1.2. Нулевое поколение (механические вычислители)

К этому поколению относят механические вычислители. Первым из них является изобретенный за полвека до нашей эры *абак* (счеты), выполненный на основе примитивных каменных

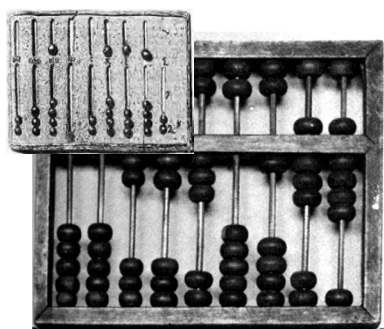


Рис. 1.1. Первые вычислители (абак)

«процессоров» – счетных камней, размещавшихся на разрядных линейках или, позднее, костяшек, нанизанных на стержни (рис. 1.1). Стержни представляли собой разряды системы счисления: каждая костяшка на первом стержне имела достоинство 1, на втором – 10 и т.д.

Знаменательным событием в области усовершенствования инструментального счета было изобретение логарифмов. В 1614 г. шотландский математик Джон Непер (1550 – 1617 гг.) опубликовал трактат «Описание удивительных таблиц логарифмов» – первое руководство по вычислениям с помощью логарифмов.

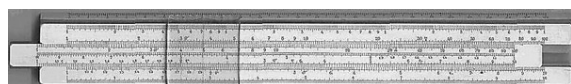


Рис. 1.2. Логарифмическая линейка

Открытие логарифмов послужило основой для создания логарифмической линейки (рис. 1.2), появление которой относят к началу XVII в. Первые логарифмические линейки были изобретены в Англии. Почти 3,5 столетия логарифмическая линейка господствовала среди всех счетных средств.

Биографии механических вычислительных машин ведутся от машины восемнадцатилетнего французского математика и физика Блеза Паскаля (1623 – 1662 гг.). Первую модель вычислительной машины, которая могла выполнять арифметические операции сложения и вычитания, он создал в 1642 г. В 1645 г. арифметическая машина «Паскалина», или «Паскалево колесо», получила законченный вид (рис. 1.3). До настоящего времени сохранилось восемь машин Паскаля. Одна из них находится в Музее искусств и ремесел в Париже, где собрана полная коллекция математических инструментов.

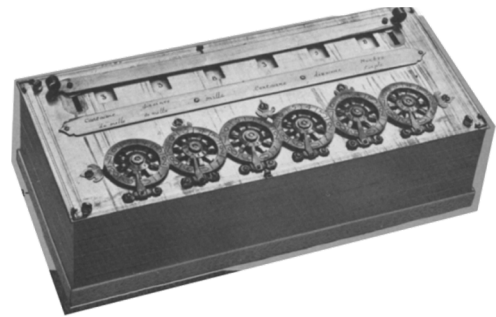


Рис. 1.3. Паскалево колесо

В 1673 г. Готфрид Вильгельм Лейбниц создал новый тип арифмометра – механический калькулятор (рис. 1.4), который не только складывал и вычитал числа, но и умножал, делил, возводил в степень, извлекал квадратные и кубические корни. В нем использовалась двоичная система счисления.

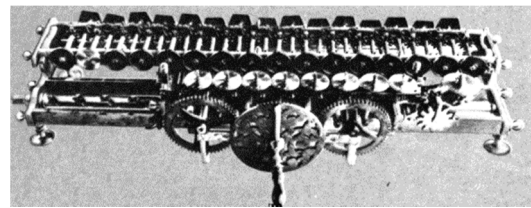


Рис. 1.4. Механический калькулятор

Самым значительным событием XIX в. в области создания вычислительной техники стал проект разностной машины английского математика Чарльза Бэббиджа (1791 – 1871 гг.), впервые в истории высказавшего идею создания вычислительных машин с программным управлением (рис. 1.5). Работать над машиной Ч. Бэббидж начал в 1812 г., к 1822 г. он построил небольшую действующую разностную машину и рассчитал на ней таблицу квадратов. Но более совершенную машину изготовить не удалось, поскольку в то время развитие техники находилось на недостаточно высоком уровне.

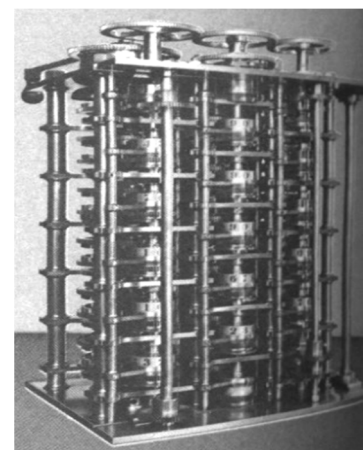


Рис. 1.5. Разностная машина Чарльза Бэббиджа

В 1841 г. занялась изучением аналитической машины Ч. Бэббиджа Ада Августа Байрон (1815 – 1852 гг.), по мужу А. Лавлейс (рис. 1.6). Лавлейс разработала первые программы для аналитической машины,



Рис. 1.6. Ада Августа Байрон

заложив тем самым теоретические основы программирования. Она впервые ввела понятие цикла операции. Ей принадлежат некоторые термины, употребляемые программистами и сейчас, например рабочие ячейки.

Крупнейший русский математик и механик П. Л. Чебышев создает в 1878 г. арифмометр, выполняющий суммирование и вычитание. В 1881 г. он изобрел приставку к своему прибору для умножения и деления.

В 1880 г. петербургский инженер В. Т. Однер создает в России усовершенствованный арифмометр; в 1890 г. он налаживает массовый выпуск арифмометров, нашедших применение во всем мире (рис. 1.7).

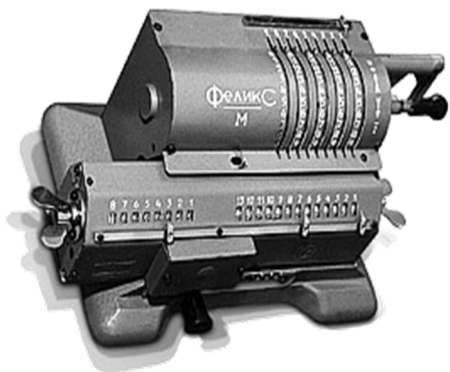


Рис. 1.7. Арифмометр Феликс

Данные в арифмометр вводились вручную, а привод осуществлялся вращением рукоятки. Их модификация «Феликс» выпускалась в СССР до 50-х гг. XX в.

Важный шаг на пути автоматизации вычислений был сделан американцем Германом Холлеритом (1860 – 1929 гг.), который изобрел электромеханические машины для вычислений с помощью перфокарт, получившие название счетно-аналитических машин.

Работая с 1882 г. в Массачусетском технологическом институте и затем в Бюро патентов США, он начал разрабатывать машины для механизации обработки данных переписи населения. В 1888 г. он создает особое устройство – табулятор, в котором информация, нанесенная на перфокарты, расшифровывается с использованием электрического тока.

В 1896 г. Г. Холлерит основал фирму по выпуску перфокарт и счетно-перфорационных машин (СПМ). В дальнейшем она была преобразована в известную фирму-производитель вычислительной техники *IBM*.

В период (1939 – 1941 гг.) доктором Конрадом Зусом был создан первый программно-управляемый компьютер Z3. Он использовал целые числа с плавающей запятой и состоял из 2600 телефонных реле.

В 1943 г. под руководством американца Говарда Айкена по заказу и при поддержке фирмы *IBM* создан *Mark-1* – программно-управляемый компьютер, также построенный на электро-механических реле (рис. 1.8), а программа обработки данных вводилась с перфоленты. Весила эта машина около 35 т.

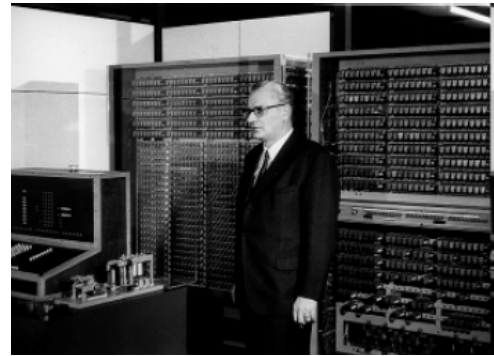


Рис. 1.8. Программно-управляемый компьютер *Mark-1*

1.3. Первое поколение (вычислители на электронных лампах)

К этому поколению относят вычислители на электронных лампах. Одной из предпосылок создания ЭВМ было изобретение лампового диода и триода. В 1904 г. Дж. Флеминг (Великобритания) изобрел первый ламповый диод, а в 1906 г. Ли де Форест и Р. Либен (США) – первый триод. Но эра ЭВМ начинается с изобретения лампового триггера. В 1918 г. русский ученый М. А. Бонч-Бруевич изобрел триггер, имевший только два устойчивых положения равновесия: «открыто», «закрыто». Это изобретение имело большое значение для создания в дальнейшем современных вычислительных машин. В 1919 г. независимо от М. А. Бонч-Бруевича такой же прибор изобрели американцы У. Икклз (Экклз) и Ф. Джордан. Триггерные схемы постепенно стали широко применяться в электронике для переключения и релейной коммутации.

В 1931 г. французский инженер Р.-Л. В. Валтат выдвинул идею использования двоичной системы счисления при создании механических счетных устройств.

В 1936 г. английский математик Алан Тьюринг и независимо от него американский математик Э. Л. Пост выдвинули и разработали концепцию абстрактной вычислительной машины.

А. Тьюринг опубликовал в 1936 г. статью в журнале Лондонского математического общества с доказательством того, что любой алгоритм

может быть реализован с помощью дискретного автомата. Он предложил схему такого автомата, получившего название машины Тьюринга (рис. 1.9).

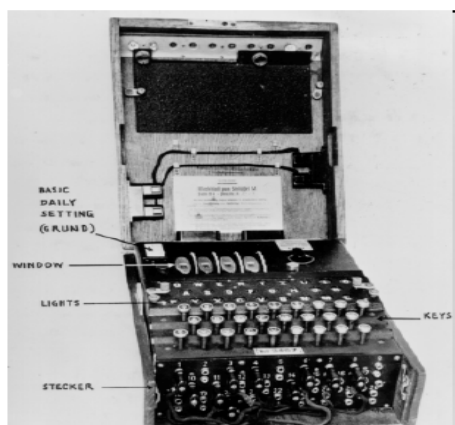


Рис. 1.9. Машина Тьюринга

Машина Тьюринга – гипотетический универсальный преобразователь дискретной информации, теоретическая вычислительная система. Работами Тьюринга теоретически была доказана возможность создания универсальной цифровой вычислительной машины.

Первыми из таких машин можно считать вычислители на электронных лампах *Enigma* (1940 г.), *Colossus-1* (1943 г.).

В 1943 г. начались работы над реализацией проекта в Пенсильванском университете под руководством Дж. Моучли и инженера-электронщика Д. П. Эккерта. Их машина, названная *ENIAC* (ЭНИАК) –

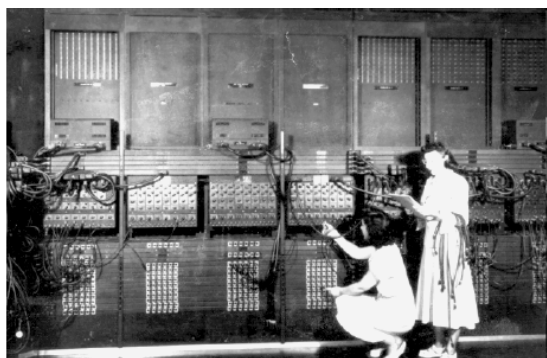


Рис. 1.10. Электронный цифровой компьютер *ENIAC*

«*Electronic Numerical Integrator and Computer*» («электронный цифровой компьютер»), в основном была закончена в 1946 г. (рис. 1.10). *ENIAC* содержал 18000 электроно-вакуумных ламп, 7200 диодов, 1500 реле, весил 30 т, потреблял 150 кВт энергии и имел время выполнения одного цикла машинной команды 200 мкс. *ENIAC* предназначался для использования при расчетах баллистических таблиц для орудий береговой обороны США. Первым найденным в США практическим применением ЭНИАК было решение задач для проекта атомной бомбы.

Первым найденным в США практическим применением ЭНИАК было решение



Рис. 1.11. ЭВМ *EDVAC*

В 1945 г. Дж. фон Нейман разработал концепцию электронно-вычислительной машины *EDVAC* (*Electronic Discrete Variable Computer*) с вводимыми в память программами и числами. Сама машина была завершена в 1950 г. (рис. 1.11). Главными элементами концепции были: принцип хра-

нимой программы и принцип параллельной организации вычислений, согласно которому операции над числом проводятся по всем его разрядам одновременно. Запущена машина в эксплуатацию была лишь в 1952 г.

Первые ЭВМ в нашей стране создавались для решения сложных и трудоемких математических задач. Работы над первой ЭВМ начались в 1946 г. под руководством академика С. А. Лебедева в Институте электротехники АН УССР. В 1947 г. был закончен проект малой электронной счетной машины – МЭСМ (рис. 1.12).



Рис. 1.12. Машина – МЭСМ

Основные опытно-конструкторские работы, монтаж и испытания прошли в 1949 – 1950 гг. В октябре 1951 г. машина была введена в эксплуатацию. Это была самая быстродействующая тогда машина в Европе. С ее помощью был решен ряд важных задач, в том числе расчет устойчивости магистральной линии электропередачи Куйбышев – Москва (1951 г.). Функциональная схема машины удовлетворяла всем принципам Дж. фон Неймана.

В 1952 г. вступил в строй опытный образец машины БЭСМ (большая электронно-счетная машина), созданный в Институте точной механики и вычисли-

тельной техники под руководством С. А. Лебедева. БЭСМ-2 (рис. 1.13) имела оперативную память на электронно-акустиче-



Рис. 1.13. Машина БЭСМ-2

ских линиях задержки, затем – на электронно-лучевых трубках и позже – на ферритовых сердечниках. Внешнее запоминающее устройство состояло из магнитных барабанов и магнитной ленты.

Серийное производство ЭВМ в нашей стране началось в 1953 г. Под руководством Ю. Я. Базилевского и Б. И. Рамеева была разработана машина «Стрела», выпускавшаяся малой серией. В 1955 г. в Пензе



Рис. 1.14. Машина – «Урал-1»

начался выпуск малой ЭВМ «Урал-1» (рис. 1.14), созданной также под руководством Б. И. Рамеева.

В 50-е гг. также разработаны серийные ЭВМ: в Ереване под руководством Ф. Т. Саркисяна – «Раздан», в Минске (В. В. Пржиялковский и др.) – ЭВМ «Минск», а в Киеве – «Киев». В 1958 г. самой мощной в Европе была советская ЭВМ М-20, созданная объединенными усилиями коллективов С. А. Лебедева и Ю. Я. Базилевского.

Характерные черты ЭВМ первого поколения:

- элементная база: электронно-вакуумные лампы, резисторы, конденсаторы. Соединение элементов – навесной монтаж проводами;
- габаритные размеры: ЭВМ выполнена в виде громоздких шкафов и занимает специальный машинный зал;
- быстродействие: 10 – 20 тыс. оп/с;
- эксплуатация: слишком сложна из-за частого выхода ЭВМ из строя;
- программирование: трудоемкий процесс в машинных кодах. Общение с ЭВМ требовало от специалистов высокого профессионализма.

1.4. Второе поколение (компьютеры на транзисторах)

Это конец 50-х – начало 60-х гг. XX в. Использование транзисторов позволило уменьшить размеры и потребляемую мощность устройств. Первый компьютер на транзисторах был произведен в США в 1954 – 1957 гг. Он имел: память 4096 слов по 18 бит, производительность 100000 оп/с, рабочую частоту 200 кГц.

Коммерческий мини-компьютер (*PDP-8 DEC*) был создан в 1964 г. (рис. 1.15). Он имел: память 4096 слов по 12 бит, производительность 333000 ком/с.

В 1974 г. появляются суперкомпьютеры линии. Суперкомпьютер *Seymour Gray (Gray-1)* (рис. 1.16) имел тактовую частоту 80 МГц, разрядность 64 бита, память 1 млн слов, потребляемую мощность 115 кВт, вес 5,5 т.

Получает развитие мультипрограммирование – совмещение во времени выполнения различных программ, а также переход от машинных к алгоритмическим языкам. К 1967 г. было уже создано около 1000 таких языков. В 1953 – 57 гг. в США (Дж. Бейкус) разработан язык *Fortran* – «переводчик формул», а в 1957 г. – алгоритмический язык *Algol*. В конце 50-х гг. Дж. Маккарти в Массачусетском технологическом институте (МТИ, США) разрабатывает *LISP* для работ по проблеме искусственного интеллекта. В 1960 г. в США создают *COBOL* – язык для обработки коммерческой информации.

Это привело к созданию в 1961 г. ЦВМ *B5000 (Burroughs Large System)*, использующей языки *Algol* и *COBOL*.

Первые ЦВМ на полупроводниках в СССР появились с 1961 г.: для научных расчетов – ЭВМ средней мощности: М-220, БЭСМ-3, БЭСМ-4 (Москва), Урал-11, -14, -16 (Пенза), Минск-22, -23, -32 (Минск), Раздан-2, -3 (Ереван), в 1961 г. – первая в стране серийная универсальная полупроводниковая ЭВМ Днепр-1.

Принципы совместимости машин (однотипность архитектуры) и агрегатной комплектации были реализованы на серии «Урал» раньше, чем внедрены на *IBM-360*.

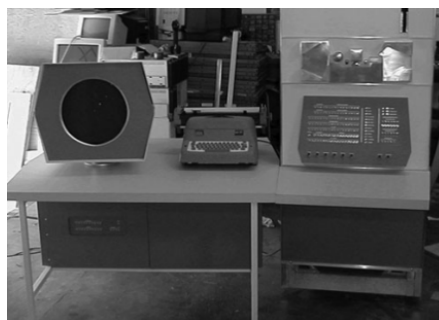


Рис. 1.15. Мини-компьютер *PDP-8 DEC*

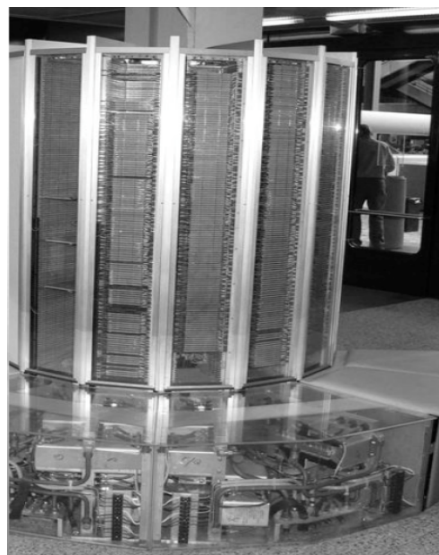


Рис. 1.16. Суперкомпьютер *Gray-1*

Лучшей в мире ЭВМ второго поколения, уровень которой на несколько лет опередил уровень зарубежных аналогичных ЭВМ, стала



Рис. 1.17. ЭВМ БЭСМ-6

БЭСМ-6 (рис. 1.17), созданная в Институте точной механики и вычислительной техники АН СССР под руководством С. А. Лебедева и В. А. Мельникова. БЭСМ-6 работала в мультипрограммном режиме. Обладая высоким быстродействием (около 1 млн операций в секунду), БЭСМ-6 по своей архитектуре была ближе к ЭВМ третьего поколения.

Характерные черты ЭВМ второго поколения:

- элементная база: полупроводниковые элементы; соединение элементов – печатные платы и навесной монтаж;
- габаритные размеры: ЭВМ выполнены в виде однотипных стоек, для их размещения требуется специально оборудованный машинный зал;
- быстродействие: до 1 млн оп/с;
- эксплуатация: при выходе из строя нескольких элементов производилась замена целиком всей платы, а не каждого элемента в отдельности, как в ЭВМ предыдущего поколения;
- программирование: велось преимущественно на алгоритмических языках в пакетном режиме.

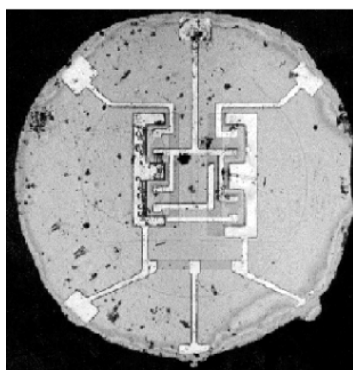


Рис. 1.18. ИС на подложке

1.5. Третье поколение (компьютеры на интегральных схемах)

Идея создания интегральных схем (ИС) была предложена в 1952 г. инженером из Великобритании Дж. Даммером. В 1959 г. Роберт Нойс (будущий основатель фирмы *Intel*) изобрел метод, позволивший создавать на одной пластине и транзисторы, и все необходимые соединения между ними (рис. 1.18). Полученные электронные схемы и стали называться ИС, или чипами.

В 1961 г. в продажу поступила первая, выполненная на пластине кремния ИС на шести элементах, в 1963 г. ИС имела 10 – 20 элементов, в 1967 г. – примерно 100, к 1970 г. – 1000, к 1975 г. – 30000, к 1982 г. – 300000 элементов на кристалле в несколько квадратных миллиметров. В 1968 г. фирма *Burroughs* выпустила первый компьютер на ИС, а в 1970 г. фирма *Intel* начала продавать ИС памяти.

Начало периода машин третьего поколения связано с разработкой ЭВМ серии *IBM-360*, позднее – *IBM-370* (рис. 1.19), оказавшей огромное влияние на развитие вычислительной техники во всем мире. Выпуск машин *IBM-360* был начат в США в 1965 г. Сущность идей, заложенных в проект *IBM-360*, – создание семейства машин на ИС, имеющих широкий диапазон производительности и совместимых на уровне машинных языков, периферийных устройств, модулей конструкции и системы элементов, а также операционных систем (ОС).



Рис. 1.19. ЭВМ *IBM-370*

Значительным явлением в этот период было создание мини-ЭВМ. Сущность идеи состояла в такой минимизации аппаратуры центрального вычислителя, которая позволяла создать универсальные ЭВМ, способные осуществлять управление в реальном масштабе времени. Первая модель ДВК-1 была разработана в 1981 г. в НИИТТ НПО «Научный Центр», г. Зеленоград (рис. 1.20).



Рис. 1.20. ДВК-1

Характеристики ДВК-1:

- процессор на основе микропроцессора *K1801BM1*;
- объём ОЗУ – 48 Кбайт;
- быстродействие – около 330 тыс. оп/с.

К третьему поколению ЭВМ принадлежат малые вычислительные машины СМ – СМ-1, СМ-2, СМ-3 и СМ-4 (рис. 1.21).



Наличие ЗУ различных емкости и быстродействия, стандартизация подключения к процессору устройств ввода-вывода символьной и графической информации открыли возможности использования СМ ЭВМ в различных измерительных, управляющих, информационных комплексах. Они сопрягались с большими вычислительными машинами как периферийные процессоры, удаленные терминалы, процессоры ввода-вывода, связи ЭВМ с объектом в реальном масштабе времени.

В это время в СССР выпускаются мини-ЭВМ Мир-31, Мир-32, Наир-34, ЭВМ серии АСВТ М-6000 и М-7000 для управления технологическими процессами; на ИС – настольные мини-ЭВМ М-180 «Электроника-100, -200», «Электроника ДЗ-28», «Электроника НЦ-60» и др.

В этот же период развитие ЭВМ в СССР проходило в направлении создания Единой системы электронных вычислительных машин (ЕС ЭВМ) (рис. 1.22), в основных чертах копирующих *IBM-360* и *IBM-370*, программно-совместимых между собой, а также с ЭВМ типа *IBM-360*. Это большие вычислительные машины серий 1020, 1030, 1040, 1060 и так далее, имеющих разрядную сетку 8 бит, набор из 144 команд, ОЗУ



Рис. 1.22. ЕС ЭВМ

до 256 Кбайт, диски емкостью 7,25 Мбайт. Они потребляли 21 кВт энергии и занимали площадь более 100 кв. м.

Характерные черты ЭВМ третьего поколения:

- элементная база – ИС;
- габаритные размеры: для размещения ЕС ЭВМ требуется машинный зал. А малые ЭВМ – это в основном две стойки, дисплей;
- быстродействие: до миллиона операций в секунду;
- эксплуатация: более оперативно производится ремонт стандартных неисправностей;
- технология программирования и решения задач: режим пакетной обработки;
- изменения в структуре ЭВМ. Наряду с микропрограммным способом управления используются принципы модульности и магистральности. Под магистральностью понимается способ связи между модулями компьютера, когда все входные и выходные устройства подсоединены одними и теми же проводами (шинами). Это прообраз современной системной шины;
- увеличились объемы памяти. Появились дисплеи, графопостроители.

1.6. Четвертое поколение (компьютеры на СБИС)

Переход к машинам четвертого поколения – ЭВМ на сверхбольших интегральных схемах (СБИС) происходил в середине и второй половине 70-х гг. и завершился приблизительно к 1980 г. Машины этого поколения развивались, во-первых, в направлении создания мощных многопроцессорных систем, имеющих производительность до сотен миллионов операций в секунду, и, во-вторых, в направлении создания дешевых компактных мини- и микроЭВМ.

Большие ЭВМ четвертого поколения представляют собой многопользовательские машины с развитыми возможностями для работы с базами данных, с различными формами удаленного доступа. Они имеют несколько процессоров и ориентированы на выполнение определенных операций, процедур или на решение некоторых классов задач.

Серия машин *IBM S/390* (рис. 1.23) продолжила линию машин *IBM-360* и *IBM-370*. Они позволяют задавать переменную конфигурацию (число процессоров – 1 – 10, емкость оперативной памяти – 512 – 81292 Мбайта, число каналов – 3 – 256).



Рис. 1.23. *IBM S/390*

машины предполагалось применять для обеспечения научных исследований.



Рис. 1.24. *RS/6000*

Они могут использоваться в качестве основы серверов в локальных сетях, для управления сложными технологическими производственными процессами.

В этот период стали серийно производиться суперЭВМ. Первый суперкомпьютер – *Cray-1* (проект С. Крея), установленный в Лос-Аламосе (США). Он выполнял до 100 млн арифметических операций в секунду.

К числу машин четвертого поколения относятся и машины *RS/6000* (рис. 1.24), предназначенные для создания графических рабочих станций, *Unix*-серверов, кластерных комплексов. Первоначально эти

К числу средних ЭВМ четвертого поколения относят ЭВМ типа *AS/400* (*Advanced Portable Model 3*) – 64-разрядные «бизнес-компьютеры». Они предназначены в первую очередь для работы в финансовых структурах. В этих машинах особое внимание уделяется сохранению и безопасности данных, программной совместимости и т.д.

Они могут использоваться в качестве основы серверов в локальных сетях, для управления сложными технологическими производственными процессами.

В нашей стране похожие характеристики имела суперЭВМ «Эльбрус-2» (рис. 1.25). Многопроцессорные комплексы «Эльбрус-2» и «Эльбрус-3» имели быстродействие соответственно 100 млн и 1 млрд оп/с. Нашими учеными была создана многопроцессорная система ПС-2000, в которой могло достигаться быстродействие до 200 млн оп/с.



Рис. 1.25. Супер-ЭВМ «Эльбрус-2»

К суперЭВМ 4-го поколения относят *GF-11 (Gigaflop-11)* с быстродействием 11 млрд оп/с (рис. 1.26).

Предварительные расчеты показали, что применение этой системы позволит решить целый комплекс новых задач. Одной из таких задач было уточнение массы протона. При использовании новой ЭВМ эта работа могла быть выполнена за 1,5 – 4 месяца.

Самый мощный в мире суперкомпьютер *K* расположен в Институте физико-химических исследований (*RIKEN*) в японском городе Кобе (рис. 1.27).



Рис. 1.26. СуперЭВМ *Gigaflop-11*

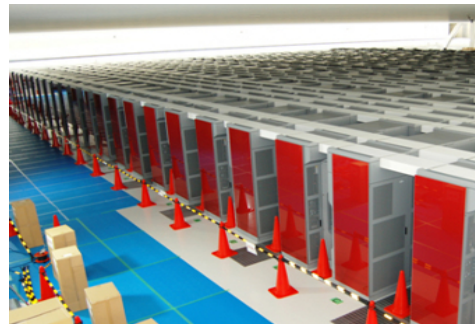


Рис. 1.27. Суперкомпьютер *K*

Конфигурация суперкомпьютера *K* объединяет 864 вычислительные стойки и 705 тысяч процессорных ядер архитектуры *Sparc64*. Теоретическая производительность системы составляет 11,28 петафлопс (ПФ = 10^{15} оп/с). Он будет применяться для прогнозирования стихийных бедствий, в частности землетрясений, для анализа процессов изменения климата и в космических исследованиях.

Персональные компьютеры

Выпуск первого микропроцессора *Intel-4004* (4 бита) служит датой начала микропроцессорной революции. Она шла двумя путями: МП встраиваются в аппаратуру (связи, производственную, медицинскую, бытовую и т. д.) или служат основой построения соответствующих управляющих устройств, в частности персональных ЭВМ (ПЭВМ).



Рис. 1.28. ПК «Альтаир-8800»

В начале 1975 г. появился первый коммерчески распространяемый ПК «Альтаир-8800» (рис. 1.28) фирмы *MITS* (США), построенный на основе МП *Intel-8080*.

В конце 1975 г. Пол Аллен и Билл Гейтс (будущие основатели фирмы *Microsoft*) создали для компьютера «Альтаир» интерпретатор языка *Basic*, что позволило пользователям просто общаться с компьютером и легко писать для него программы.

Революция в индустрии ПК была совершена фирмами *IBM* и *Apple Computer*. При этом первым всенародным ПК стал компьютер фирмы *Apple Computer* (рис. 1.29). История этой фирмы началась в 1976 г., когда Стивен Джобс и Стивен Возник создали свою первую модель, назвав ее «*Apple*» (яблоко).



Рис. 1.29. ПК фирмы *Apple Computer*

Развитием работ в данном направлении стал компьютер *Apple II*, созданный на 8-рядном МП фирмы *Motorola*.

В 1981 г. на рынке ПК появилась фирма *IBM*, которая во много раз превосходила по своим возможностям фирму *Apple*. В качестве основного МП компьютера *IBM* был выбран *Intel-8088*. В данном компьютере были использованы и другие комплектующие различных фирм, а его ПО поручили разработать небольшой фирме *Microsoft*.

Основой популярности *IBM PC* является заложенная в нем возможность усовершенствования его отдельных частей и использования новых устройств. Фирма *IBM* сделала его не единым неразъемным устройством, а обеспечила возможность сборки из частей, изготовленных независимо различными фирмами.

В 1983 г. был выпущен *IBM PC XT* (*Personal Computer Extended Technology*), имеющий встроенный жесткий диск, в 1985 г. *IBM PC AT* (*Personal Computer Advanced Technology*) на основе нового МП *Intel-80286* (рис. 1.30), работающий в 3 – 4 раза быстрее *IBM PC XT*.



Рис. 1.30. ПК *IBM PC AT*

Принцип открытой архитектуры лишил фирму *IBM* монополии на выпуск ПК класса *IBM PC*. Наибольшее влияние на развитие компьютеров типа *IBM PC* теперь оказывает не *IBM*, а фирма *Intel* – производитель МП и фирма *Microsoft* – разработчик ОС *Windows* и многих других используемых на *IBM PC* программ.

Что же касается *Apple Computer*, то она сохранила особенности своей модели, отличной от *IBM PC*. Компьютер, выполненный фирмой на самом новом в то время микропроцессоре *Motorola 68000*, получил имя *Macintosh* (рис. 1.31) – название одного из популярных сортов американских яблок «макинтош».



Рис. 1.31. ПК *Macintosh*

В начале 1984 г. *Macintosh* поступил в продажу. В комплекте с новейшим лазерным принтером *Macintosh* с его графическими возможностями стал идеальной машиной для развивающихся в то время настольных издательств. Полюбился компьютер и преподавателям школ, колледжей, университетов. А вот в деловом мире по-прежнему популярностью пользовались *IBM*-совместимые компьютеры.

В нашей стране в 1988 г. был начат массовый выпуск школьных персональных компьютеров и классов учебной вычислительной тех-

ники (КУВТ) «Корвет», «Электроника УКНЦ» и других, профессиональных компьютеров ДВК-2 (рис. 1.32), ДВК-3М, ДВК-4, Искра-1030, Нейрон, ЕС-1841 и др.



Рис. 1.32. ПК ДВК-2

В настоящее время на компьютерном рынке рядом с фирмами *IBM* и *Apple Computer* трудятся такие крупные фирмы, как *Compaq*, *Packard Bell*, *Dell*, *Hewlett-Packard* и др.

В 1979 г. по заказу *NASA* Вильям Моргридж (компания *Grid Systems*) создал первый в мире ноутбук *Grid Compass* (оперативная память на ЦМД объемом 340 Кб, процессор *Intel 8086* с тактовой частотой 8 МГц).

Первый ноутбук (лэптоп) фирмой *IBM* (*IBM Convertible*) был выпущен 3 апреля 1986 г., он был снабжен дисковыми для 3,5" дискет, подобно современным ноутбукам был способен работать от батарей и имел функции управления питанием.



Рис. 1.33. Электроника МС-1504

«Электроника МС 1504» – первый советский персональный компьютер в форм-факторе «*laptop*» на процессоре *KP1834BM86* (1992 г.) (рис. 1.33). Первоначально имел название ПК-300 и цену 550 дол. США. В качестве прототипа был использован лэптоп «*T1100 PLUS*» фирмы *Toshiba*.

Разновидность персональных компьютеров – планшетный ПК (или *tablet PC*) (рис. 1.34) – появилась в широкой

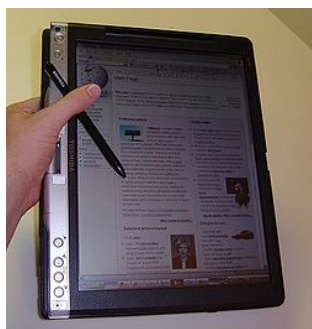


Рис. 1.34. Планшетный ПК

продаже после презентации аппаратно-программной платформы *Microsoft Tablet PC*, разработанной компанией *Microsoft* и представленной 7 ноября 2002 г. До этого времени устройства такого типа использовались на более малых рынках – на производстве, в медицине и госучреждениях. Планшетные ПК и сегодня широко используются в госучреждениях и корпоративной среде.

Главная отличительная особенность данного семейства ПК – это аппаратная совместимость с *IBM PC*-компьютерами и установленные на них полноценные операционные системы, используемые на настольных компьютерах и ноутбуках.

Характерные черты ЭВМ четвертого поколения:

- элементная база: СБИС;
- габаритные размеры: для больших ЭВМ и суперЭВМ требуется машинный зал, малые ЭВМ – это в основном стойки, персональные компьютеры – размещаются на столе пользователя;
- производительность: десятки-сотни миллионов оп/с;
- организация: многопользовательские, многопроцессорные машины с развитыми возможностями для работы с базами данных, различными формами удаленного доступа;
- эксплуатация: совместимость программного обеспечения снизу вверх и принцип открытой архитектуры, предусматривающий возможность дополнения имеющихся аппаратных средств без смены старых или их модификации без замены всего компьютера.

1.7. Пятое поколение (компьютеры с искусственным интеллектом)

С конца 80-х гг. XX в. в истории развития ЭВМ наступила пора пятого поколения машин. Проект машин пятого поколения разработан ведущими японскими фирмами и научными организациями, согласно этому проекту ЭВМ и вычислительные системы коренным образом отличаются от машин предшествующих поколений.

Прежде всего, их структура отличается от той, которую предложил Джон фон Нейман, и содержит:

- блок общения, обеспечивающий интерфейс между пользователем и ЭВМ на языке, близком к естественному;
- базу знаний, хранящую все необходимые для решения задач сведения о той предметной области, к которой эти задачи относятся;

- решатель, который организует подготовку программы решения задачи на основании знаний из базы знаний и исходных данных, полученных из блока общения.

ЭВМ пятого поколения должны самоорганизовываться в процессе решения задач, иметь собственную внутреннюю модель мира и активно взаимодействовать с внешней средой, а также распознавать образы, делать выводы из информации, уметь оперировать в нечетких ситуациях, пополнять имеющиеся знания (т.е. иметь способность обучаться), вести диалог с человеком на естественном речевом или графическом языке, иметь способность понимать содержимое базы знаний и использовать эти знания при решении задач.

Таким образом, в этих машинах широко используются модели и средства, разработанные в искусственном интеллекте. В них, в частности,



Рис. 1.35. ПК пятого поколения *PIM/m-1*

широко используются языки, характерные для представления знаний, модели знаний в виде семантических сетей, фреймов и продукций. На рис. 1.35 изображен компьютер пятого поколения *PIM/m-1*, один из немногих, увидевших свет (Япония).

Разработка машин пятого поколения ведется на основе сверхбольших интегральных схем (СБИС), а также на основе перехода к супермикроэлектронике, где расстояния между элементами схем будут меньше микрона, и на основе использования достижений интегральной микрооптоэлектроники, в которой каналами связи являются световые лучи, а для преобразования электрических сигналов в световые и наоборот используются лазеры, свето- и фотодиоды. Ведутся разработки по замене кремния и биологическими системами памяти, и датчиками. Благодаря достижениям современной техники только в

компьютерах пятого поколения стало возможным технически реализовать идеи, высказанные еще в 70-е гг. Подробнее с материалом можно ознакомиться в [10], [17].

Отличительные черты ЭВМ пятого поколения:

- новая технология производства, возможно, не на кремнии, а на базе новых материалов;
- отказ от традиционных языков высокого уровня (Фортран, Паскаль и др.) в пользу языков с повышенными возможностями манипулирования символами и с элементами логического программирования (Пролог, ЛИСП);
- акцент на новые архитектуры (например, на архитектуру потока данных) и отход от структур фон Неймана;
- новые способы ввода-вывода, удобные для пользователя (например, распознавание речи и образов, синтез речи, обработка сообщений на естественном языке);
- искусственный интеллект, тесно связанный с исследованиями в области экспертных систем.

Вопросы к компьютерному тестированию

1. Как называют инструмент вычислений на основе счетных камней, размещавшихся на разрядных линейках?
2. Кто впервые в истории высказал идею создания вычислительных машин с программным управлением?
3. Кто впервые разработал программы для аналитической машины, заложив тем самым теоретические основы программирования?
4. Кто впервые разработал арифмометр, выполнявший суммирование и вычитание?
5. Кто впервые изобрел электромеханические машины для вычислений с помощью перфокарт?
6. Кто является основоположником эры ЭВМ, начавшейся с момента изобретения лампового триггера?

7. Кем впервые была доказана возможность создания универсальной цифровой вычислительной машины?
8. Кто впервые разработал концепцию (архитектуру) электронно-вычислительной машины?
9. Кто является родоначальником первых отечественных больших вычислительных машин?
10. К какому времени относятся разработка и производство первых в мире компьютеров на транзисторах?
11. К какому времени относятся разработка и производство первых отечественных компьютеров на транзисторах?
12. К какому времени относятся разработка и производство первых суперкомпьютеров?
13. К какому периоду времени относится разработка первых алгоритмических языков?
14. Кому принадлежит идея создания интегральных схем, положившая начало ЭВМ третьего поколения?
15. К какому поколению ЭВМ принадлежат малые вычислительные машины серии СМ?
16. Компьютеры какого поколения как минимум требуются для создания графических рабочих станций, *Unix*-серверов, кластерных комплексов?
17. К какому периоду времени относится переход к машинам четвертого поколения?
18. Какое название имели первые отечественные суперкомпьютеры, быстродействие которых превышало 100 млн оп/с?
19. Какую производительность (оп/с) имеет наиболее мощный из известных в настоящее время суперкомпьютеров?
20. К какому времени относятся разработка и производство первых коммерчески распространяемых ПК?
21. Какое название имел первый коммерчески распространяемый ПК?
22. Какое название имел первый всенародный ПК?
23. Какую разрядность процессора имел первый всенародный ПК?

24. К какому времени относятся разработка и производство первых массовых персональных компьютеров производства фирмы *IBM*?
25. К какому времени относятся разработка и производство первых в мире ноутбуков?
26. К какому времени относятся разработка и производство первых отечественных ноутбуков?
27. К какому времени относятся разработка и производство первых планшетных ПК?
28. Какое название имел первый отечественный ноутбук?
29. Назовите основные особенности ЭВМ 5-го поколения.
30. Что прежде всего отличает структуру ЭВМ 5-го поколения от той, которую предложил Джон фон Нейман?

Проверка усвоения изложенного в лекции материала проводится по приведенным вопросам с помощью тестирующей программы, осуществляющей подготовку к поблочному рейтинговому контролю знаний в виде контрольного тестирования.

Подготовительное тестирование выполняется с использованием режима тренинга, где без ограничения по времени последовательно предлагаются все 30 вопросов каждой из лекций и даются подсказки о правильности введенных ответов.

Контрольное тестирование осуществляется с использованием тестов, прилагаемых в конце каждого из трех блоков лекционного материала, включающих по три лекции (т.е. выполняется в конце 3-й, 6-й и 9-й лекций).

Оно также запускается из главного меню и предлагает 10 вопросов с глобальным ограничением по времени 10 мин. Имеется возможность пропуска вопросов, вызывающих затруднение, и возвращение к ним в конце теста. Результат тестирования оценивается по 500-балльной системе.



Глава 2

АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ

Рассматриваемые вопросы:

- 2.1. Системы счисления.
- 2.2. Перевод целых чисел.
- 2.3. Перевод дробных чисел.
- 2.4. Представление информации в ЭВМ.
 - 2.4.1. Представление числовой информации.
 - 2.4.2. Представление других видов информации.
- 2.5. Кодирование информации.
- 2.6. Выполнение арифметических операций.
 - 2.6.1. Машинные коды.
 - 2.6.2. Арифметические операции над числами с фиксированной точкой.
 - 2.6.3. Арифметические операции над двоичными числами с плавающей точкой.
 - 2.6.4. Арифметические операции над двоично-десятичными кодами чисел.

2.1. Системы счисления

Система счисления – это способ изображения чисел с помощью ограниченного набора символов, имеющих определенные количественные значения.

По-другому можно сказать, что это совокупность приемов и правил, по которым числа записываются и читаются.

Существуют непозиционные и позиционные системы счисления.

В непозиционных системах счисления каждое число обозначается соответствующей совокупностью символов, вес цифры *не зависит от ее позиции* в записи числа. Так, в римской системе счисления в числе XXXII (тридцать два) вес цифры X в любой позиции равен просто десяти.

В позиционных системах каждая цифра числа имеет определенный вес, зависящий от позиции цифры в последовательности, изображающей число, например, числа 127 и 721. Позиция цифры называется *разрядом*. В позиционной системе счисления любое число можно представить в виде

$$A_n = a_{n-1} N^{n-1} + a_{n-2} N^{n-2} + \dots + a_1 N^1 + a_0 N^0 + a_{-1} N^{-1} + \dots + a_{-m} N^{-m},$$

или $A_n = \sum_{i=-m}^{n-1} a_i N^i$,

где a_i – цифры системы счисления; n и m – число целых и дробных разрядов, N – основание системы счисления, соответственно.

Основание позиционной системы счисления – количество различных цифр, используемых для изображения чисел в данной системе счисления.

За основание системы можно принять любое натуральное число – два, три, четыре и так далее, следовательно, возможно бесчисленное множество позиционных систем: двоичная, троичная, четверичная и т.д.

Пример 2.1

Запись двоичного числа 1011,1 и восьмеричного числа 276,52:

Разряды		3	2	1	0	-1
Число	1 0 1 1,	$1 \cdot 2^3$	$+ 0 \cdot 2^2$	$+ 1 \cdot 2^1$	$+ 1 \cdot 2^0$	$+ 1 \cdot 2^{-1}$;
Разряды		2	1	0	-1	-2
Число	2 7 6, 5	$2 \cdot 8^2$	$+ 7 \cdot 8^1$	$+ 6 \cdot 8^0$	$+ 5 \cdot 8^{-1}$	$+ 2 \cdot 8^{-2}$.

Теоретически наиболее экономичной системой счисления является система с основанием $e = 2,71828\dots$, находящимся между числами 2 и 3.

Во всех современных ЭВМ для представления числовой информации используется двоичная система счисления. Это обусловлено:

- более простой реализацией алгоритмов выполнения арифметических и логических операций;
- более надежной физической реализацией основных функций, так как они имеют всего два состояния (0 и 1);
- экономичностью аппаратной реализации всех схем ЭВМ.

При $N = 2$ число различных цифр, используемых для записи чисел, ограничено множеством из двух цифр (нуль и единица). Кроме двоичной системы счисления широкое распространение получили и производные системы:

- двоично-десятичное представление десятичных чисел – $\{0, 1, \dots, 9\}$;
- шестнадцатеричная – $\{0, 1, 2, \dots, 9, A, B, C, D, E, F\}$. Здесь шестнадцатеричная цифра A обозначает число 10, B – число 11, ..., F – число 15;
- восьмеричная (от слова *восьмерик*) – $\{0, 1, 2, 3, 4, 5, 6, 7\}$. Она широко используется во многих специализированных ЭВМ.

Восьмеричная и шестнадцатеричная системы счисления являются производными от двоичной, так как $16 = 2^4$ и $8 = 2^3$. Они используются в основном для более компактного изображения двоичной информации, так как запись значения чисел производится существенно меньшим числом знаков.

Представление чисел в различных системах счисления допускает однозначное преобразование их из одной системы в другую. В ЭВМ перевод из одной системы в другую осуществляется автоматически по специальным программам. Правила перевода целых и дробных чисел отличаются.

2.2. Перевод целых чисел

Целое число с основанием N_1 переводится в систему счисления с основанием N_2 путем последовательного деления числа A_n на основание N_2 , записанное в виде числа с основанием N_1 , до получения остатка. Полученное частное следует вновь делить на основание N_2 , и этот процесс надо повторять до тех пор, пока частное не станет меньше делителя. Полученные остатки от деления и последнее частное записываются в порядке, обратном полученному при делении. Сформированное число и будет числом с основанием N_2 .

Пример 2.2

$$A_{10} = 37; A_2 = ?; A_{16} = ?$$

$$1) \begin{array}{r|l} 37 & 2 \\ \hline \end{array}$$

$$\begin{array}{r|l} 1 \ 18 & 2 \\ \hline 0 \ 9 & 2 \\ \hline 1 \ 4 & 2 \\ \hline 0 \ 2 & 2 \\ \hline 0 & 1 \end{array}$$

$$A_{10} = 37;$$

$$A_2 = 100101;$$

$$2) \begin{array}{r|l} 37 & 16 \\ \hline 5 & 2 \end{array}$$

$$A_{10} = 37;$$

$$A_{16} = 25.$$

2.3. Перевод дробных чисел

Дробное число с основанием N_1 переводится в систему счисления с основанием N_2 путем последовательного умножения A_n на основание N_2 , записанное в виде числа с основанием N_1 . При каждом умножении целая часть произведения берется в виде очередной цифры соответствующего разряда, а оставшаяся дробная часть принимается за новое множимое. Число умножений определяет разрядность полученного результата, представляющего число A_n , в системе счисления N_2 .

У чисел, содержащих целую и дробную части, перевод осуществляется отдельно по каждой из частей.

Пример 2.3.

$$A_{10} = 0,625; \quad A_2 = ? \quad A_8 = ? \quad A_{16} = ?$$

<p>a)</p> $\begin{array}{r} 0,625 \\ \times 2 \\ \hline 1\,250 \\ \times 2 \\ \hline 0\,500 \\ \times 2 \\ \hline 1\,000 \end{array}$ <p>$A_2 = 0,101$</p>	<p>b)</p> $\begin{array}{r} 0,625 \\ \times 8 \\ \hline 5\,000 \end{array}$ <p>$A_8 = 0,5$</p>	<p>c)</p> $\begin{array}{r} 0,625 \\ \times 16 \\ \hline 10\,000 \end{array}$ <p>$A_{16} = 0,4$</p>
---	---	--

Пример 2.4

$$A_{10} = 1,375; \quad A_2 = ?$$

Целая часть $1_{10} = 1_2$, дробная часть $0,375$

$$A_{2 \text{ цел}} = 1$$

$$A_{2 \text{ дробн}} = 0,011$$

$$A_2 = 1,011$$

$$\begin{array}{r} 0,375 \\ \times 2 \\ \hline 0\,750 \\ \times 2 \\ \hline 1\,500 \\ \times 2 \\ \hline 1\,000 \end{array}$$

Так как двоичная, восьмеричная и шестнадцатеричная системы связаны через степени числа 2, то преобразования между ними можно выполнять другим более простым способом. Для перевода из шестнадцатеричной (восьмеричной) системы счисления в двоичную достаточно двоичным кодом записать шестнадцатеричные коды цифр тетрадами (по 4 двоичных разряда) и триадами (по 3 двоичных разряда) – для восьмеричных цифр.

Обратный перевод из двоичного кода производится в обратном порядке: двоичное число разбивается влево и вправо от границы целой и дробной частей на тетрады – для последующей записи цифр в шестнадцатеричном представлении, на триады – для записи их значений восьмеричными цифрами.

2.4. Представление информации в ЭВМ

Информация – это сведения об окружающем мире и протекающих в нем процессах, воспринимаемые человеком или специализированным устройством, например ЭВМ, для обеспечения целенаправленной деятельности.

Информация может быть по своей физической природе: *числовой, текстовой, графической, звуковой, видео и др.* Она также может быть постоянной (неменяющейся), переменной, случайной, вероятностной. Существуют различные способы оценки количества информации. Классическим является подход, использующий формулу К. Шеннона. Применительно к двоичной системе она имеет вид

$$H = \log_2 N,$$

где H – количество информации, несущей представление о состоянии, в котором находится объект;

N – количество равновероятных альтернативных состояний объекта.

Любая информация, обрабатываемая в ЭВМ, должна быть представлена двоичными цифрами {0, 1}, т.е. должна быть закодирована комбинацией этих цифр. Различные виды информации (числа, тексты, графика, звук) имеют свои правила кодирования. Коды отдельных значений, относящиеся к различным видам информации, могут совпадать. Поэтому расшифровка кодированных данных осуществляется по контексту при выполнении команд программы.

2.4.1. Представление числовой информации

В ЭВМ используются три вида чисел: *с фиксированной точкой (запятой), с плавающей точкой (запятой) и двоично-десятичное представление.* Точка (запятая) – это подразумеваемая граница целой и дробной частей числа.

У чисел *с фиксированной точкой* в двоичном формате предполагается строго определенное место точки. Обычно это место определяется или перед первой значащей цифрой числа, или после последней

значащей цифры числа. Если точка фиксируется перед первой значащей цифрой, то это означает, что число по модулю меньше единицы.

Если точка фиксируется после последней значащей цифры, то это означает, что n -разрядные двоичные числа являются целыми.

Перед самым старшим из возможных разрядов двоичного числа фиксируется его знак. Положительные числа имеют нулевое значение знакового разряда, отрицательные – единичные.

Другая форма представления чисел – представление их в виде чисел с плавающей точкой (запятой). *Числа с плавающей точкой представляются в виде мантиссы m_a и порядка p_a* , иногда это представление называют полулогарифмической формой числа.

Например, число $A_{10} = 373$ можно представить в виде $0,373 \cdot 10^3$, при этом $m_a = 0,373$, $p_a = 3$, основание системы счисления подразумевается фиксированным и равным десяти. Для двоичных чисел A_2 в этом представлении также формируется $+/-m_a$ и порядок $+/-p_a$ при основании системы счисления, равном двум, что соответствует записи $A_2 = 2^{+/-p_a}(+/-m_a)$.

Порядок числа p_a определяет положение точки (запятой) в двоичном числе. Значение порядка лежит в диапазоне $-p_a^{\max} \leq p_a \leq p_a^{\max}$, где величина p_a^{\max} определяется числом разрядов r , отведенных для представления порядка $p_a^{\max} = 2^r - 1$.

Положительные и отрицательные значения порядка значительно усложняют обработку вещественных чисел. Поэтому во многих современных ЭВМ используют не прямое значение p_a , а *модифицированное p'_a* , приведенное к интервалу $0 \leq p'_a \leq 2p_a^{\max}$.

Значение p'_a носит название «*характеристика числа*».

Обычно под порядок (модифицированный порядок или характеристику) выделяют один байт. Старший разряд характеристики отводится под знак числа, а семь оставшихся разрядов обеспечивают изменение порядка в диапазоне $-64 \leq p_a \leq 63$.

Модифицированный порядок p'_a вычисляется по зависимости $p'_a = p_a + 64$.

Этим самым значения p'_a формируются в диапазоне положительных чисел $0 \leq p'_a \leq 127$.

Мантисса числа m_a представляется двоичным числом, у которого точка фиксируется перед старшим разрядом.

Если старший значащий разряд мантииссы в системе счисления с основанием N отличен от нуля, то такое число называется нормализованным. Например, $A_2 = (100; 0,101101)_2$ – нормализованное число $A_2 = 1011,01$, а то же самое число $A_2 = (101; 0,0101101)_2$ – число ненормализованное, так как старший разряд мантииссы равен нулю.

Третья форма представления двоичных чисел – **двоично-десятичная**. Она используется при обработке больших массивов десятичных чисел (например, больших экономических документов). При этом каждая цифра десятичного числа представляется двоичной тетрадой. Например, $A_{10} = 3759$, $A_{2-10} = 0011\ 0111\ 0101\ 1001$. Положение десятичной запятой (точки), отделяющей целую часть от дробной, обычно заранее фиксируется. Значение знака числа отмечается кодом, отличным от кодов цифр. Например, «+» имеет значение тетрады «1100», а «-» – «1101».

2.4.2. Представление других видов информации

Различные виды информации могут быть разделены на две группы: *статические и динамические*. Так, числовая, логическая и символьная информация является статической – ее значение не связано со временем. В отличие от перечисленных типов вся *аудиоинформация* имеет динамический характер. Она существует только в режиме реального времени, ее нельзя остановить для более подробного изучения. Если изменить масштаб времени (увеличить или уменьшить), аудиоинформация искажается. Это свойство иногда используется для получения звуковых эффектов.

Видеоинформация может быть как статической, так и динамической.

Статическая видеоинформация включает текст, рисунки, графики, чертежи, таблицы и др. Рисунки делятся также на плоские – двумерные и объемные – трехмерные.

Динамическая видеоинформация – это видео-, мульт- и слайд-фильмы. В их основе лежит последовательное экспонирование на экране в реальном масштабе времени отдельных кадров в соответствии со сценарием.

По способу формирования видеоизображения бывают *растровые, матричные и векторные*.

Растровые видеоизображения используются в телевидении, а в ЭВМ практически не применяются.

Матричные изображения получили в ЭВМ наиболее широкое распространение. Изображение на экране рисуется электронным лучом точками, каждая из которых – пиксел (*picture element*) – рассматривается как наименьшая структурная единица изображения.

Векторные изображения состояются из отрезков линий (в простейшем случае – прямых), для которых задаются: начальные координаты, угол наклона и длина отрезка (может указываться и код используемой линии). Векторный способ имеет ряд преимуществ перед матричным: изображение легко масштабируется с сохранением формы, является «прозрачным», может быть наложено на любой фон и т.д.

2.5. Кодирование информации

Для кодирования *символьной* или *текстовой информации* применяются различные системы: при вводе информации с клавиатуры кодирование происходит при нажатии клавиши, на которой изображен требуемый символ, при этом в клавиатуре вырабатывается так называемый *scan-код*, представляющий собой двоичное число, равное порядковому номеру клавиши.

Номер нажатой клавиши никак не связан с формой символа, нанесенного на клавише. Оpozнание символа и присвоение ему внутреннего кода ЭВМ производятся специальной программой по специальным таблицам: *ДКОИ*, *КОИ-7*, *ASCII (American Standard Code for Information Interchange)*.

Американский стандартный код передачи информации (*ASCII*) представлен табл. 2.1.

Табл. 2.1. Американский стандартный код передачи информации

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
0.	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1.	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2.		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3.	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4.	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5.	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6.	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7.	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Этот код ставит в соответствие каждому символу 7-разрядный двоичный код ($2^7 = 128$ символов). Например, 16-ричный код символа *A* запишется *41H* (его двоично-десятичный 100 0001).

Здесь кодирование десятичных цифр строится по весовому принципу, согласно которому двоичный код символа возрастает последовательно на 1 при переходе от цифры к цифре (от 0 до 9) и от буквы к букве (в алфавитном порядке).

Такое построение кода значительно упрощает наиболее частые операции обработки текстов – сортировку и поиск. Первые две колонки в таблице содержат управляющие символы, используемые для

Табл. 2.2. Расширенный код *ASCII*

	0	1	2	3	...	7	8	9	A	B	...	E	F
0													
1													
2													
.													
.	Зона стандарта <i>ASCII</i>						Зона дополнительных символов						
7													
8													
9													
A													
B													
C													
D													
E													
F													

передачи команд.

Расширенный код *ASCII*, полностью включающий стандарт *ASCII* и дополнительно 128 символов (кодов с единицей в старшем бите), представлен в табл. 2.2.

Среди дополнительных символов – буквы греческого алфавита, математические символы, символы псевдографики и др.









Для представления символов кириллицы В. Брябриным и А. Чижовым была предложена *альтернативная кодировка* (табл. 2.3), в которой русские буквы занимают столбцы 8, 9, A, E и частично F, замещая буквы европейских алфавитов и математических символов

(псевдографика оставлена – поэтому англоязычные программные продукты функционируют нормально).

Нижняя часть таблиц кодировки (латиница) полностью соответствует кодировке *ASCII*. В приведённых таблицах числа под буквами обозначают шестнадцатеричный код буквы в Юникоде.

В машинах класса *AT* и старше альтернативную кодировку сменил ее модифицированный вариант, где изменен столбец *F* с целью большей совместимости с *IBM* таблицей, но в результате исключена буква Ё и несколько других символов, включая стрелки. Существуют и другие кодировки, например *16-битная кодировка Unicode*.

Табл. 2.3. Альтернативная кодировка

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	А 410	Б 411	В 412	Г 413	Д 414	Е 415	Ж 416	З 417	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E	П 41F
9.	Р 420	С 421	Т 422	У 423	Ф 424	Х 425	Ц 426	Ч 427	Ш 428	Щ 429	Ъ 42A	Ы 42B	Ь 42C	Э 42D	Ю 42E	Я 42F
A.	а 430	б 431	в 432	г 433	д 434	е 435	ж 436	з 437	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E	п 43F
B.	 2591	 2592	 2593	 2502	┆ 2524	≡ 2561	≡ 2562	≡ 2556	≡ 2555	≡ 2563	≡ 2551	≡ 2557	≡ 255D	≡ 255C	≡ 255B	≡ 2510
C.	┆ 2514	┆ 2534	┆ 252C	┆ 251C	— 2500	┆ 253C	┆ 255E	┆ 255F	┆ 255A	┆ 2554	┆ 2569	┆ 2566	┆ 2560	┆ 2550	┆ 256C	┆ 2567
D.	┆ 2568	┆ 2564	┆ 2565	┆ 2559	┆ 2558	┆ 2552	┆ 2553	┆ 256B	┆ 256A	┆ 2518	┆ 250C	 2588	 2584	 258C	 2590	 2580
E.	р 440	с 441	т 442	у 443	ф 444	х 445	ц 446	ч 447	ш 448	щ 449	ъ 44A	ы 44B	ь 44C	э 44D	ю 44E	я 44F
F.	Ё 401	ё 451	Є 404	є 454	İ 407	ı 457	ÿ 40E	ÿ 45E	° B0	· 2219	· B7	√ 221A	№ 2116	¤ A4	■ 25A0	А0

Описание формы каждого символа хранится в специальной памяти дисплея – знакогенераторе.

Высвечивание символа на экране дисплея *IBM PC* осуществляется с помощью точек, образующих символьную матрицу.

Каждый пиксел в такой матрице – элемент изображения и может быть ярким или темным. Темная точка кодируется цифрой 0, светлая (яркая) – 1.

Если изображать в матричном поле знака темные пиксели точкой, а светлые – звездочкой, то можно графически изобразить форму символа.

Кодирование аудиоинформации – процесс более сложный. Аудиоинформация является аналоговой. Для преобразования ее в цифровую форму используют аппаратные средства: *аналого-цифровые преобразователи (АЦП)*, в результате работы которых аналоговый сигнал оцифровывается – представляется в виде числовой последовательно-

сти. Для вывода оцифрованного звука на *аудиоустройства* необходимо проводить обратное преобразование, которое выполняется с помощью *цифро-аналоговых* преобразователей (ЦАП).

2.6. Выполнение арифметических операций

Все современные ЭВМ имеют достаточно развитую систему команд, включающую десятки и сотни машинных операций. Однако выполнение любой операции основано на использовании простейших микроопераций типа сложения и сдвига. Это позволяет иметь единое арифметико-логическое устройство для выполнения любых операций, связанных с обработкой информации. Правила сложения двоичных цифр двух чисел A и B представлены в табл. 2.4.

Здесь показаны правила сложения двоичных цифр a_i , b_i одноименных разрядов с учетом возможных переносов из предыдущего разряда P_{i-1} .

Подобные таблицы можно было бы построить для любой другой арифметической и логической операции (вычитание, умножение и т.д.), но именно данные этой таблицы положены в основу выполнения любой операции ЭВМ. Под знак чисел отводится специальный знаковый разряд. Знак «+» кодируется двоичным нулем, а знак «-» – единицей. Действия над прямыми кодами двоичных чисел при выполнении операций создают большие трудности, связанные с необходимостью учета значений знаковых разрядов:

- во-первых, следует отдельно обрабатывать значащие разряды чисел и разряды знака;
- во-вторых, значение разряда знака влияет на алгоритм выполнения операции (сложение может заменяться вычитанием и наоборот).

Табл. 2.4. Правила сложения двоичных цифр

Значения двоичных чисел A и B			Разряд суммы S_i	Перенос в следующий разряд P_i
a_i	b_i	p_{i-1}		
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Во всех ЭВМ без исключения все операции выполняются над числами, представленными специальными машинными кодами. Их использование позволяет обрабатывать знаковые разряды чисел так же, как и значащие разряды, а также заменять операцию вычитания операцией сложения.

Различают прямой код (П), обратный код (ОК) и дополнительный код (ДК) двоичных чисел.

2.6.1. Машинные коды

Прямой код двоичного числа образуется из абсолютного значения этого числа и кода знака (ноль или единица) перед его старшим числовым разрядом.

Пример 2.5

$A_{10} = +10; A_2 = +1010; \text{Код: } [A_2]_{\text{п}} = 0: 1010;$

$B_{10} = -15; B_2 = -1111; \text{Код: } [B_2]_{\text{п}} = 1: 1111.$

Двоеточием здесь отмечена условная граница, отделяющая знаковый разряд от значащих.

Обратный код двоичного числа образуется по следующему правилу. Обратный код положительных чисел совпадает с их прямым кодом. Обратный код отрицательного числа содержит единицу в знаковом разряде числа, а значащие разряды числа заменяются на инверсные, т.е. нули заменяются единицами, а единицы – нулями.

Пример 2.6

$A_{10} = +5; A_2 = +101; [A_2]_{\text{п}} = [A_2]_{\text{ок}} = 0: 101;$

$B_{10} = -13; B_2 = -1101; [B_2]_{\text{ок}} = 1: 0010.$

Свое название обратный код чисел получил потому, что коды цифр отрицательного числа заменены на инверсные. Укажем наиболее важные свойства обратного кода чисел:

- сложение положительного числа C с его отрицательным значением в обратном коде дает так называемую машинную единицу $ME_{\text{ок}} = 1: 111...11$, состоящую из единиц в знаковом и значащих разрядах числа;

- нуль в обратном коде имеет двойное значение. Он может быть положительным – $0: 00...0$ и отрицательным числом – $1: 11...11$. Значение отрицательного нуля совпадает с $ME_{\text{ок}}$. Двойственное представление нуля явилось причиной того, что в современных ЭВМ все числа представляются не обратным, а дополнительным кодом.

Дополнительный код положительных чисел совпадает с их прямым кодом. Дополнительный код отрицательного числа представляет собой результат суммирования обратного кода числа с единицей младшего разряда (2^0 – для целых чисел, 2^{-K} – для дробных).

Пример 2.7

$A_{10} = +19; A_2 = +10011; [A_2]_{\text{п}} = [A_2]_{\text{ок}} = [A_2]_{\text{дк}} = 0: 10011;$
 $B_{10} = -13; B_2 = -1101; [B_2]_{\text{дк}} = [B_2]_{\text{ок}} + 2^0 = 1: 0010 + 1 = 1: 0011.$

Основные свойства дополнительного кода:

- сложение дополнительных кодов положительного числа C с его отрицательным значением дает так называемую машинную единицу дополнительного кода:

$$\text{МЕ}_{\text{дк}} = \text{МЕ}_{\text{ок}} + 2^0 = 10: 00\dots 00,$$

т.е. число 10 (два) в знаковых разрядах числа;

- дополнительный код получил такое свое название потому, что представление отрицательных чисел является дополнением прямого кода чисел до машинной единицы $\text{МЕ}_{\text{дк}}$.

Модифицированные обратные и дополнительные коды двоичных чисел отличаются соответственно от обратных и дополнительных кодов удвоением значений знаковых разрядов. Знак «+» в этих кодах кодируется двумя нулевыми знаковыми разрядами, а «-» – двумя единичными разрядами.

Пример 2.8

$A_{10} = 9; A_2 = +1001; [A_2]_{\text{п}} = [A_2]_{\text{ок}} = [A_2]_{\text{дк}} = 0: 1001.$

$[A_2]_{\text{мок}} = [A_2]_{\text{мдк}} = 00: 1001;$

$B_{10} = -9; B_2 = -1001; [B_2]_{\text{ок}} = 1: 0110; [B_2]_{\text{дк}} = 1: 0111;$

$[B_2]_{\text{мок}} = 11: 0110; [B_2]_{\text{мдк}} = 11: 0111.$

Целью введения модифицированных кодов являются фиксация и обнаружение случаев получения неправильного результата, когда значение результата превышает максимально возможный результат в отведенной разрядной сетке машины. В этом случае перенос из значащего разряда может исказить значение младшего знакового разряда. Значение знаковых разрядов «01» свидетельствует о положительном переполнении разрядной сетки, а «10» – об отрицательном переполнении. В настоящее время практически во всех моделях ЭВМ роль удвоенных разрядов для фиксации переполнения разрядной сетки играют переносы, идущие в знаковый и из знакового разряда.

2.6.2. Арифметические операции над числами с фиксированной точкой

Сложение (вычитание). Операция вычитания приводится к операции сложения путем преобразования чисел в обратный или дополнительный код. Пусть числа $A \geq 0$ и $B \geq 0$, тогда операция алгебраического сложения выполняется в соответствии с табл. 2.5.

Табл. 2.5. Таблица преобразования кодов при алгебраическом сложении

Требуемая операция	Необходимое преобразование
$A + B$	$A + B$
$A - B$	$A + (-B)$
$-A + B$	$(-A) + B$
$-A - B$	$(-A) + (-B)$

Скобки в представленных выражениях указывают на замену операции вычитания операцией сложения с обратным или дополнительным кодом соответствующего числа. Сложение двоичных чисел осуществляется последовательно, поразрядно в соответствии с табл. 2.4. При выполнении сложения цифр необходимо соблюдать следующие правила.

1. Слагаемые должны иметь одинаковое число разрядов. Для выравнивания разрядной сетки слагаемых можно дописывать незначащие нули слева к целой части числа и незначащие нули справа к дробной части числа.

2. Знаковые разряды чисел участвуют в сложении так же, как и значащие.

3. Необходимые преобразования кодов (п. 2.4.1) производятся с изменением знаков чисел. Приписанные незначащие нули изменяют свое значение при преобразованиях по общему правилу.

4. При образовании единицы переноса из старшего знакового разряда в случае использования ОК эта единица складывается с младшим числовым разрядом. При использовании ДК единица переноса теряется. Знак результата формируется автоматически, результат представляется в том коде, в котором представлены исходные слагаемые.

Пример 2.9

Сложить два числа $A_{10} = 7$; $B_{10} = 16$.

$$A_2 = +111; \quad B_2 = +10000.$$

Исходные числа имеют различную разрядность, необходимо провести выравнивание разрядной сетки:

$$[A_2]_{\text{п}} = [A_2]_{\text{ок}} = [A_2]_{\text{дк}} = 0: 00111;$$

$$[B_2]_{\text{п}} = [B_2]_{\text{ок}} = [B_2]_{\text{дк}} = 0: 10000.$$

Сложение в обратном или дополнительном коде дает один и тот же результат

$$\begin{array}{r} 0: 00111 \\ + 0: 10000 \\ \hline C_2 = 0: 10111 \\ C_{10} = +23. \end{array}$$

Обратим внимание, что при сложении цифр отсутствуют переносы в знаковый разряд и из знакового разряда, что свидетельствует о получении правильного результата.

Пример 2.10

Сложить два числа $A_{10} = +16$; $B_{10} = -7$ в ОК и ДК. В соответствии с табл. 2.5 должна быть реализована зависимость $A + (-B)$, в которой второй член преобразуется с учетом знака.

$$[A_2]_{\text{п}} = 0 : 10000; [A_2]_{\text{ок}} = 0 : 10000; [A_2]_{\text{дк}} = 0 : 10000;$$

$$[B_2]_{\text{п}} = 1 : 111 = 1 : 00111; [B_2]_{\text{ок}} = 1 : 11000; [B_2]_{\text{дк}} = 1 : 11001.$$

Сложение в ОК

$$\begin{array}{r} [A_2]_{\text{ок}} = 0 : 10000 \\ + [B_2]_{\text{ок}} = 1 : 11000 \\ \hline \end{array}$$

$$\begin{array}{r} \begin{array}{c} \curvearrowright \quad \curvearrowright \\ 0 : 01000 \end{array} \\ \hline 1 \\ 0 : 01001 \end{array}$$

$$C_2 = 0 : 01001;$$

$$C_{10} = +9;$$

Сложение в ДК

$$\begin{array}{r} [A_2]_{\text{дк}} = 0 : 10000 \\ + [B_2]_{\text{дк}} = 1 : 11001 \\ \hline \end{array}$$

$$\begin{array}{r} \begin{array}{c} \curvearrowright \quad \curvearrowright \\ 0 : 01001 \end{array} \end{array}$$

$$C_2 = 0 : 01001;$$

$$C_{10} = +9.$$

При сложении чисел в ОК и ДК были получены переносы в знаковый разряд и из знакового разряда. В случае ОК перенос из знакового разряда требует дополнительного прибавления единицы младшего разряда (см. п. 4 правил). В случае ДК этот перенос игнорируется.

Умножение. Умножение двоичных чисел наиболее просто реализуется в прямом коде. Рассмотрим, каким образом оно приводится к операциям сложения и сдвигам.

Пример 2.11

Умножить два числа $A_{10} = 7$; $B_{10} = 5$.

Перемножим эти числа, представляемые прямыми двоичными кодами, так же, как это делается в десятичной системе.

$$\begin{array}{r}
 [A_2]_{\text{п}} = 111 \text{ – множимое} \\
 \times \\
 [B_2]_{\text{п}} = \underline{101} \text{ – множитель} \\
 \quad 111 \text{ – множимое} \qquad \qquad \qquad \text{(сдвиг на 0 разрядов)} \\
 \quad + 000 \text{ – умножение на 0} \qquad \qquad \text{(сдвиг на 1 разряд)} \\
 \quad \underline{111} \text{ – множимое} \qquad \qquad \qquad \text{(сдвиг на 2 разряда)} \\
 [C_2]_{\text{п}} = 100011 \\
 C_{10} = 35.
 \end{array}$$

Нетрудно видеть, что произведение получается путем сложения частных произведений, представляющих собой разряды множимого, сдвинутые влево в соответствии с позициями разрядов множителя. Частные произведения, полученные умножением на нуль, игнорируются. Важная особенность операции умножения n -разрядных сомножителей – увеличение разрядности произведения до $n + n = 2n$. Знак произведения формируется путем сложения знаковых разрядов сомножителей. Возможные переносы из знакового разряда игнорируются.

Деление. Операция деления, как и в десятичной арифметике, является обратной операции умножения. Покажем, что и эта операция приводится к последовательности операций сложения и сдвига.

Пример 2.12

Разделим два числа $A_{10} = 45$; $B_{10} = 5$.

$$\begin{array}{l}
 [A_2]_{\text{п}} = 101101; \\
 [B_2]_{\text{п}} = 101;
 \end{array}$$

Делимое	Делитель
101101	<u>101</u>
<u>101</u>	1001 – частное
0101	
<u>101</u>	
0	

[C₂]_п = 1001;
C₁₀ = 9.

Деление произведено так же, как это делается обычно в десятичной системе. Сначала проверяется, можно ли вычесть значение делителя из старших разрядов делимого. Если возможно, то в разряде частного записывается единица и определяется частная разность. В противном случае в частное записывается нуль и разряды делителя сдвигаются вправо на один разряд по отношению к разрядам делимого. К полученной предыдущей разности сносится очередная цифра делимого, и данный процесс повторяется, пока не будет получена необходимая точность. Если учесть, что все вычитания в ЭВМ заменяются сложением в ОК или ДК (см. табл. 2.4), то действительно операция деления приводится к операциям сложения и сдвигам вправо разрядов делителя относительно разрядов делимого. Отметим, что делимое перед операцией деления должно быть приведено к $2n$ -разрядной сетке. Только в этом случае при делении на n -разрядный делитель получается n -разрядное частное.

Знак частного формируется также путем сложения знаковых разрядов делимого и делителя, как это делалось при умножении.

2.6.3. Арифметические операции над двоичными числами с плавающей точкой

В современных ЭВМ числа с плавающей точкой хранятся в памяти машин, имея мантиссу и порядок (характеристику) в прямом коде и нормализованном виде. Все арифметические действия над этими числами выполняются так же, как это делается с ними, если они представлены в полулогарифмической форме (мантисса и десятичный порядок) в десятичной системе счисления. Порядки и мантиссы обрабатываются раздельно.

Сложение (вычитание). Операция сложения (вычитания) производится в определенной последовательности.

1. Сравниваются порядки (характеристики) исходных чисел путем их вычитания $\Delta p = p_1 - p_2$. При выполнении этой операции определяется, одинаковый ли порядок имеют исходные слагаемые.

2. Если разность порядков равна нулю, то это значит, что одноименные разряды мантисс имеют одинаковые веса (двоичный порядок). В противном случае должно проводиться выравнивание порядков.

3. Для выравнивания порядков число с меньшим порядком сдвигается вправо на разницу порядков Δp . Младшие выталкиваемые разряды при этом теряются.

4. После выравнивания порядков мантиссы чисел можно складывать (вычитать) в зависимости от требуемой операции. Операция вычитания заменяется операцией сложения в соответствии с данными табл. 2.5. Действия над слагаемыми производятся в ОК или ДК по общим правилам.

5. Порядок результата берется равным большему порядку.

6. Если мантисса результата не нормализована, то выполняются нормализация и коррекция значений порядка.

Пример 2.13

Сложить два числа $A_{10} = +1,375$; $B_{10} = -0,625$.

$A_2 = +1,011 = 0: 1011 \cdot 10^1$; $B_2 = -0,101 = -0: 101 \cdot 10^0$.

В нормализованном виде эти числа выглядят следующим образом:

Порядок	Мантисса
$[A_2]n = 0: 1$	0: 1011
	} знак числа
$[B_2]n = 0: 0$	1: 101

1. Вычитаем порядки $\Delta p = p_1 - p_2 = 1 - 0 = 1$. В машине эта операция требует операции сложения с преобразованием порядка чисел в дополнительный код:

$$\begin{array}{rcl}
 p_1 = 0: 1 & & [p_1]_{\text{дк}} = 0: 1 \\
 p_2 = 0: 0 & + & [p_2]_{\text{дк}} = 0: 0 \\
 & & \hline
 & & \Delta p = 0: 1
 \end{array}$$

Определяем, что $\Delta p \neq 0$.

2. Порядок первого числа больше порядка второго числа на единицу. Требуется выравнивание порядков.

3. Для выравнивания порядков необходимо второе число сдвинуть вправо на один разряд.

$$[B_2]_{\text{исх}} = 0:0 \ 1:101.$$

$$[B_2]_{\text{исх}} = 0:0 \ 1:101.$$

$$\text{После сдвига: } [B'_2]_n = 0: 11: 0101;$$

$$\text{После сдвига: } [m'_B]_{\text{дк}} = 1: 1011.$$

4. Складываем мантииссы.

$$[m_A]_{\text{дк}} = 0: 1011$$

$$\underline{[m'_B]_{\text{дк}} = 1: 1011}$$

$$[m_C]_{\text{дк}} = 0: 0110.$$

Здесь мантиисса числа B в дополнительном коде $(m_B)_{\text{дк}}$ образуется из мантииссы этого числа в обратном коде (1010) с добавлением 1 в младшем разряде (1011).

Мантиисса числа C – положительная.

5. Порядок числа C равен порядку числа с большим порядком, т.е. $p_C = +1$.

$$[C_2]_n = 0: 1 \ 0: 0110.$$

Видно, что мантиисса результата не нормализована, так как старшая цифра мантииссы равна нулю.

6. Нормализуем результат путем сдвига мантииссы на один разряд влево и соответственно вычитаем из значения порядка единицу:

$$[C_2]_n = 0: 0 \ 0: 110;$$

$$C_{10} = +0,75.$$

Умножение (деление). Операция умножения (деления) чисел с плавающей точкой также требует разных действий над порядками и мантииссами. Алгоритмы этих операций выполняются в определенной последовательности.

1. При умножении (делении) порядки складываются (вычитаются) так, как это делается над числами с фиксированной точкой.

2. При умножении (делении) мантииссы перемножаются (делятся).

3. Знаки произведения (частного) формируются путем сложения знаковых разрядов сомножителей (делимого и делителя). Возможные переносы из знакового разряда игнорируются.

2.6.4. Арифметические операции над двоично-десятичными кодами чисел

При обработке больших массивов экономической информации переводы чисел из десятичной системы в двоичную и обратно могут требовать значительного машинного времени. Некоторые образцы ЭВМ поэтому имеют или встроенные, или подключаемые блоки, которые обрабатывают десятичные целые числа в их двоично-десятичном представлении. Действия над ними также приводятся к операции алгебраического сложения отдельных цифр чисел, представленных дополнительными кодами в соответствии с табл. 2.5.

Приведем один из алгоритмов сложения, который получил довольно широкое распространение.

1. Сложение чисел начинается с младших цифр (тетрад) и производится с учетом возникающих переносов из младших разрядов в старшие.

2. Знак суммы формируется специальной логической схемой по знаку большего слагаемого.

3. Для того чтобы при сложении двоично-десятичных цифр возникали переносы, аналогичные при сложении чисел в десятичном представлении, необходимо проводить так называемую десятичную коррекцию. Для этого к каждой тетраде первого числа прибавляется дополнительно по цифре $b_{10} = 0110_2$, что позволяет исключить шесть неиспользуемых комбинаций $(1010 - 1111)_2$, так как они кодируют шестнадцатеричные цифры $A - F$ (числа $10 - 15_{10}$).

4. После операции суммирования проводится корректировка суммы. Из тех тетрад суммы, из которых не было переносов, изымаются ранее внесенные избытки $b_{10} = 0110_2$. Для этого проводится вторая коррекция. Операция вычитания заменяется, как и обычно, операцией сложения с числом -6 , представленным дополнительным кодом 1010_2 , но только в тех разрядах, в которых отсутствовали переносы. При этой второй коррекции переносы из тетрад блокируются.

5. Операция вычитания реализуется достаточно своеобразно. По общему правилу сложения (п. 1 – 4) к тетрадам числа с большим модулем прибавляются дополнительные коды тетрад другого числа. В качестве знака результата берется знак числа с большим модулем.

Пример 2.14

Сложить два числа $A_{10} = 177$; $B_{10} = 418$

A_{2-10}	00001	0111	0111	—	—	—	—	—	—	
+										
A'	0110	0110	0110							
	0111	1101	1101							
+										
B_{2-10}	0100	0001	1000	—	—	—	—	—	—	
	1011	1111	0101							
+										
C_{2-10}	1010	1010								
	0101	1001	0101							

$C_{10}=595$

Вопросы к компьютерному тестированию

1. Как называют систему счисления, в которой каждое число обозначается соответствующей совокупностью символов?
2. Как называют систему счисления, в которой каждая цифра числа имеет определенный вес?
3. Привести формулу для представления любого числа в позиционной системе счисления.
4. При каком основании система счисления является теоретически наиболее экономичной?
5. Какое число в десятичной системе счисления обозначает цифра D в шестнадцатеричной системе счисления?
6. Запишите формулу Шеннона применительно к двоичной системе счисления.
7. Как обозначаются двоичные числа с фиксированной точкой по модулю меньше единицы?
8. Какое значение принимает знаковый разряд у отрицательных чисел с фиксированной точкой?
9. Запишите численное значение порядка числа 25,3 при записи его в виде нормализованного числа с плавающей запятой.
10. Запишите численное значение мантиссы числа 25,3 при записи его в виде нормализованного числа с плавающей запятой.

11. Какой объем памяти ЭВМ отводится под порядок нормализованного числа с плавающей запятой?
12. Какие из известных видов информации являются динамическими?
13. Какие из видов изображения рисуются электронным лучом точками?
14. Какие из видов изображения состояются из отрезков линий?
15. Как называют код, представляющий собой двоичное число, равное порядковому номеру клавиши?
16. Запишите число, равное разрядности расширенного кода *ASCII*?
17. Какие из операций над числами выполняют с помощью арифметико-логического устройства ЭВМ?
18. Какое значение (1 или 0) принимает разряд суммы двоичного сумматора при единичных входных сигналах и нулевом переносе из предыдущего разряда?
19. Приведите запись числа (6) в обратном машинном коде.
20. Приведите запись числа (–6) в прямом машинном коде.
21. Приведите запись числа (6) в дополнительном машинном коде.
22. Приведите запись числа (6) в обратном модифицированном машинном коде.
23. Приведите числовую запись машинной единицы.
24. Что происходит при сложении двоичных чисел и образовании единицы переноса из старшего знакового разряда в случае использования ОК?
25. Как формируется знак произведения при операции умножения n -разрядных сомножителей?
26. Как формируется порядок результата при сложении двоичных чисел с плавающей запятой?
27. Как формируется порядок результата при умножении двоичных чисел с плавающей запятой?
28. Учитываются ли возможные переносы из знакового разряда при арифметических операциях над двоичными числами с плавающей запятой?
29. Как формируется знак при вычитании чисел в двоично-десятичном коде?
30. В каком случае блокируются возникающие переносы из младших разрядов в старшие при сложении чисел в двоично-десятичном коде?



Глава 3

ЛОГИЧЕСКИЕ ОСНОВЫ ЭВМ

Рассматриваемые вопросы:

- 3.1. Основные понятия алгебры логики.
- 3.2. Связь алгебры логики с системой кодирования данных ЭВМ.
- 3.3. Законы алгебры логики.
- 3.4. Понятие о минимизации логических функций.
- 3.5. Техническая интерпретация логических функций.

3.1. Основные понятия алгебры логики

Теоретическая основа построения ЭВМ – специальные математические дисциплины. Одной из них является алгебра логики или булева алгебра (по имени английского математика прошлого столетия (рис. 3.1), основоположника этой дисциплины) [19]. Ее аппарат широко используют для описания схем ЭВМ, их оптимизации и проектирования.



Рис. 3.1. Дж. Буль

Логическое высказывание – это любое повествовательное предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Так, например, предложение "*6 – четное число*" следует считать высказыванием, так как оно истинное.

Предложение "*Рим – столица Франции*" тоже высказывание, так как оно ложное.

Предложения типа "*В городе А более миллиона жителей*" не являются высказыванием, так как для выяснения его истинности или ложности нужны дополнительные сведения: о каком конкретно городе идет речь. Такие предложения называются *высказывательными формами*.

Алгебра логики рассматривает любое высказывание только с одной точки зрения – является ли оно истинным или ложным.

Употребляемые в обычной речи слова и словосочетания "не", "и", "или", "если..., то", "тогда и только тогда" и другие позволяют из уже заданных высказываний строить новые высказывания. Такие слова и словосочетания называются *логическими связками*.

Высказывания, образованные из других высказываний с помощью логических связок, называются *составными*. Высказывания, не являющиеся составными, называются *элементарными*.

Так, например, из элементарных высказываний "Петров – врач", "Петров – шахматист" при помощи связки "и" можно получить составное высказывание "Петров – врач и шахматист", понимаемое как "Петров – врач, хорошо играющий в шахматы".

Чтобы обращаться к логическим высказываниям, им назначают имена. Пусть через A обозначено высказывание "Перед нами лазерный принтер", а через B – высказывание "Принтер перед нами цветной". Тогда составное высказывание "Принтер перед нами лазерный и цветной" можно кратко записать как A и B . Здесь "и" – логическая связка, A , B – логические переменные, которые могут принимать только два значения – "истина" или "ложь", обозначаемые соответственно "1" и "0".

Каждая логическая связка рассматривается как операция над логическими высказываниями и имеет свое название и обозначение.

Операция, выражаемая словом "не", называется *отрицанием* и обозначается чертой над высказыванием (или знаком \neg).

Операция, выражаемая связкой "и", называется *конъюнкцией* (лат. *conjunctio* – соединение), или логическим умножением, и обозначается точкой "·" (может также обозначаться знаками \wedge или $\&$). Высказывание $A \cdot B$ истинно тогда и только тогда, когда оба высказывания A и B истинны.

Операция, выражаемая связкой "или" (в неисключающем смысле этого слова), называется *дизъюнкцией* (лат. *disjunctio* – разделение) или логическим сложением и обозначается знаком \vee (или плюсом). Высказывание $A \vee B$ ложно тогда и только тогда, когда оба высказывания A и B ложны.

Операция, выражаемая связками "если ..., то", "из ... следует", "... влечет ...", называется *импликацией* (лат. *implico* – тесно связаны) и обозначается знаком \rightarrow . Высказывание $A \rightarrow B$ ложно тогда и только тогда, когда A истинно, а B ложно.

Например, имеем высказывание "Саша – студент" (A) и высказывание "Все студенты сдают экзамен" (B). Рассмотрим составное высказывание $A \rightarrow B$, понимаемое как "если Саша студент, то он должен сдавать экзамен". Есть три варианта, когда высказывание $A \rightarrow B$ истинно:

1. A истинно и B истинно, т.е. *Саша – студент, и он должен сдавать экзамен*;
2. A ложно и B истинно, т.е. *Саша не студент, но он должен сдавать экзамен* (например, для получения прав);
3. A ложно и B ложно, т.е. *Саша не студент, и он не должен сдавать экзамен*.

Ложен только один вариант, когда A истинно, а B ложно, т.е. *Саша – студент, и он не должен сдавать экзамен*.

В обычной речи связка "если ..., то" описывает причинно-следственную связь между высказываниями. Но в логических операциях смысл высказываний не учитывается. Рассматривается только их истинность или ложность. Поэтому не надо удивляться "бессмысленности" импликаций, образованных высказываниями, совершенно не связанными по содержанию. Например, такими: "Если пусто в голове, то разбился стакан".

Операция, выражаемая связками "тогда и только тогда", "необходимо и достаточно", "... равносильно ...", называется эквиваленцией, или двойной импликацией, и обозначается знаком \leftrightarrow или \sim . Высказывание $A \leftrightarrow B$ истинно тогда и только тогда, когда значения A и B совпадают. Например, высказывания "18 делится на 6 тогда и только тогда, когда 18 делится на 3", "13 делится на 6 тогда и только тогда, когда 11 делится на 3" – истинны, а высказывания "18 делится на 6 тогда и только тогда, когда 18 делится на 5", "17 делится на 6 тогда и только тогда, когда 6 делится на 3" – ложны.

Итак, нами рассмотрены пять логических операций: отрицание, конъюнкция, дизъюнкция, импликация и эквиваленция.

Импликацию можно выразить через дизъюнкцию и отрицание:

$$A \rightarrow B = \bar{A} \vee B.$$

Эквиваленцию можно выразить через отрицание, дизъюнкцию и конъюнкцию:

$$A \leftrightarrow B = (\bar{A} \vee B) \cdot (\bar{B} \vee A).$$

Таким образом, операций отрицания, дизъюнкции и конъюнкции достаточно, чтобы описывать и обрабатывать логические высказывания.

Порядок выполнения логических операций задается круглыми скобками. Но для уменьшения числа скобок договорились считать, что сначала выполняется операция отрицания ("не"), затем – конъюнкции ("и"), после конъюнкции – дизъюнкции ("или") и в последнюю очередь – импликация.

3.2. Связь алгебры логики с системой кодирования данных ЭВМ

Вся информация в ЭВМ представляется в двоичной системе счисления. Поставим в соответствие входным сигналам отдельных устройств ЭВМ соответствующие значения x_i ($i = 1, n$), а выходным сигналам – значения функций y_j ($j = 1, m$) (рис. 3.2).



Рис. 3.2. Связь входных и выходных сигналов ЭВМ

В этом случае зависимостями $y_j = f(x_1, x_2, \dots, x_i, \dots, x_n)$, где x – i -й вход; n – число входов; y_j – j -й выход; m – число выходов в устройстве, можно описывать алгоритм работы любого устройства ЭВМ. Каждая такая зависимость y_j является «булевой функцией», у которой число возможных состояний и каждой ее независимой переменной равно двум, т.е. функцией алгебры логики, а ее аргументы определены на множестве $\{0, 1\}$. Алгебра логики устанавливает основные законы формирования и преобразования логических функций, позволяет представить любую сложную функцию в виде композиции простейших функций. Рассмотрим наиболее употребительные из них.

Известно, что количество всевозможных функций N от n аргументов выражается зависимостью

$$N = 2^{2^n} . \tag{3.1}$$

При $n = 0$ можно определить две основные функции ($N = 2$), не зависящие от каких-либо переменных: y_0 , тождественно равную нулю ($y_0 \equiv 0$), и y_1 , тождественно равную единице ($y_1 \equiv 1$). Технической интерпретацией функции $y_1 \equiv 1$ может быть источник единичного напряжения. При отсутствии входных сигналов на выходе этого устройства всегда имеется сигнал единицы. Функция $y_0 \equiv 0$ может быть интерпретирована отключенной схемой, сигналы от которой не поступают ни к каким устройствам.

При $n = 1$ зависимость (3.1) дает $N = 4$. Представим зависимость значений этих функций от значения аргумента x в виде специальной таблицы истинности (табл. 3.1).

Табл. 3.1. Таблица функций от одной переменной

$x_1 \backslash y_j$	y_0	y_1	y_2	y_3
0	0	1	0	1
1	0	1	1	0

Таблицы истинности получили такое название, потому что они определяют значение функции в зависимости от комбинации входных сигналов. В этой таблице, как и ранее, $y_0 \equiv 0$ и $y_1 \equiv 1$. Функция $y_2 = x$, а функция $y_3 = \bar{x}$ (инверсия x).

Этим функциям соответствуют определенные технические аналоги. Схема, реализующая зависимость $y_2 = x$, называется повторителем, а схема $y_3 = \bar{x}$ – инвертором.

При $n = 2$, $n = 16$, т.е. от двух переменных, можно построить шестнадцать различных функций. В табл. 3.2 представлена часть из них, имеющая фундаментальное значение при построении основных схем ЭВМ.

Табл. 3.2. Таблица функций от двух переменных

$x_1x_2 \backslash y_j$	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{15}
00	0	1	0	1		0	1	0	1	1	0		
01	0	1	0	1	1	0	0	1	0	1	
10	0	1	1	0		1	0	0	1	0	1		
11	0	1	1	0		1	0	1	0	1	0		

В левой части таблицы перечислены всевозможные комбинации входных переменных (наборы значений), а в правой – возможные реакции выходных сигналов. Функции y_0 - y_3 полностью соответствуют функциям табл. 3.1, а функции y_4 - y_9 – новые.

По данной таблице нетрудно составить аналитическое выражение (зависимость) для каждой функции от двух аргументов вида (3.1). Для этого наборы переменных, при которых функция принимает значение единицы, записываются как конъюнкции (их называют минтермами) и связываются знаками логического сложения. Такие формы функций получили название дизъюнктивных нормальных форм (ДНФ). Если в этих функциях конъюнкции содержат все без исключения переменные в прямом или инверсном значениях, то такая форма функций называется совершенной (СДНФ).

Если взять наборы переменных, на которых функция принимает нулевое значение, записать их как дизъюнкции (их называют макстермами) и связать знаками логического умножения, то такие формы функций получили название конъюнктивных нормальных форм (КНФ) и соответственно СКНФ.

Функция y_4 представляет собой функцию логического сложения, дизъюнкцию. Она принимает значение единицы, если значение единицы имеет хотя бы одна переменная x_1 или x_2 . СДНФ функции

$$y_4 = x_1\bar{x}_2 \vee \bar{x}_1x_2 \vee x_1x_2 = x_1 \vee x_2.$$

Соответственно СКНФ функции

$$y_4 = x_1 + x_2 = x_1 \vee x_2.$$

Тождественность приведенных аналитических зависимостей можно установить, пользуясь законами алгебры логики, приведенными ниже. Функция y_5 является инверсной функцией по отношению к y_4 :

$$y_5 = (\bar{y}_4) = \overline{(x_1 \vee x_2)} = x_1x_2.$$

Она имеет название «отрицание дизъюнкции». Иногда в литературе встречается ее специальное название «стрелка Пирса», по фамилии математика, исследовавшего ее свойства.

Функция y_6 является функцией логического умножения. Она очень похожа на операцию обычного умножения и принимает значение единицы в тех случаях, когда все ее переменные равны единице:

$$y_6 = x_1 \cdot x_2.$$

Функция y_7 является инверсной функцией по отношению к y_6 :

$$y_7 = \bar{y}_6 = \overline{x_1 \cdot x_2} = \bar{x}_1 \vee \bar{x}_2.$$

Она называется «отрицание конъюнкции», или «штрих Шеффера». Функция y_8 называется логической равнозначностью, она принимает значение единицы, если все ее переменные имеют одинаковое значение (или 0 или 1):

$$y_8 = \bar{x}_1 \cdot \bar{x}_2 \vee x_1 \cdot x_2.$$

Функция y_9 является инверсной по отношению к y_8 :

$$y_9 = \overline{y_8} = \overline{\bar{x}_1 \cdot \bar{x}_2 \vee x_1 \cdot x_2} = \bar{x}_1 \cdot x_2 \vee x_1 \cdot \bar{x}_2.$$

Она принимает значение единицы, если ее переменные имеют противоположные значения. Ниже будет показано, что функции y_8 и y_9 являются основой для построения сумматоров, так как они соответствуют правилам формирования цифр двоичных чисел при сложении (вычитании).

В качестве примера составим таблицу истинности для формулы $x \cdot y \vee \bar{x} \vee \bar{y} \vee x$, которая содержит две переменные x и y . В первых двух столбцах таблицы запишем четыре возможных пары значений этих переменных, в последующих столбцах – значения промежуточных формул и в последнем столбце – значение формулы. В результате получим табл. 3.3:

Табл. 3.3. Таблица истинности для формулы $x \cdot y \vee \bar{x} \vee \bar{y} \vee x$

Переменные		Промежуточные логические формулы					Формула
x	y	\bar{x}	$\bar{x} \cdot y$	$x \vee y$	$\bar{x} \vee \bar{y}$	$x \cdot y \vee \bar{x} \vee \bar{y}$	$x \cdot y \vee \bar{x} \vee \bar{y} \vee x$
0	0	1	0	0	1	1	1
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	1	0	0	1	0	0	1

Из таблицы видно, что при всех наборах значений переменных x и y формула принимает значение 1, т.е. является *тождественно истинной*.

Из перечисленных выше функций двух переменных можно строить сколь угодно сложные зависимости, отражающие алгоритмы преобразования информации, представленной в двоичной системе счисле-

ния. Алгебра логики устанавливает правила формирования логически полного базиса простейших функций, из которых могут строиться любые более сложные. Наиболее привычным базисом является набор трех функций {инверсия – \neg , дизъюнкция – \vee , конъюнкция – \wedge или $\&$ }. Работа с функциями, представленными в этом базисе, очень похожа на использование операций обычной алгебры.

3.3. Законы алгебры логики

Из определения вышеприведенных функций можно установить целый ряд простейших свойств:

$$\begin{aligned} x \vee 1 &= 1; & x \cdot 0 &= 0; \\ x \vee \bar{x} &= 1; & x \cdot 1 &= x; & x \vee x \vee \dots \vee x &= x; \\ x \vee 0 &= x; & x \cdot \bar{x} &= 0; & x \cdot x \cdot \dots \cdot x &= x; \\ x \vee x &= x; & x \cdot x &= x. \end{aligned}$$

В алгебре логики установлен целый ряд законов, с помощью которых возможно преобразование логических функций (ЛФ):

коммутативный (переместительный)

$$x_1 \cdot x_2 = x_2 \cdot x_1; \quad x_1 \vee x_2 = x_2 \vee x_1;$$

ассоциативный (сочетательный)

$$(x_1 \cdot x_2) \cdot x_3 = (x_1 \cdot x_3) \cdot x_2 = x_1 (x_2 \cdot x_3); \quad (x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3).$$

дистрибутивный (распределительный)

$$x_1 (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3; \quad x_1 \vee x_2 \cdot x_3 = (x_1 \vee x_2)(x_1 \vee x_3).$$

Эти законы полностью идентичны законам обычной алгебры.

Закон поглощения. В дизъюнктивной форме ЛФ конъюнкция меньшего ранга, т.е. с меньшим числом переменных, поглощает все конъюнкции большего ранга, если ее изображение содержится в них. Это же справедливо и для конъюнктивных форм:

$$x_1 \vee x_1 \cdot x_2 = x_1; \quad x_1 (x_1 \vee x_2) = x_1.$$

Законы склеивания

$$\begin{aligned} x_1 x_2 \vee x_1 \bar{x}_2 &= x_1; & (x_1 \vee x_2)(x_1 \vee \bar{x}_2) &= x_1, \\ Fx \vee F\bar{x} &= F; & (x \vee F)(\bar{x} \vee F) &= F, \end{aligned}$$

где F – логическая функция общего вида, не зависящая от переменной x .

Закон свертки

$$x \vee \bar{x} \vee F = x \vee F; \quad x(\bar{x} \vee F) = xF.$$

Правило де Моргана

$$\overline{x_1 \vee x_2} = \bar{x}_1 \cdot \bar{x}_2; \quad \overline{x_1 \cdot x_2} = \bar{x}_1 \vee \bar{x}_2.$$

Убедиться в тождественности приведенных зависимостей можно путем аналитических преобразований выражений или путем построения таблицы истинности для логических функций, находящихся в левой и правой частях.

Используя данные зависимости, можно преобразовывать исходные выражения в более простые (минимизировать их). По упрощенным выражениям можно построить техническое устройство, имеющее минимальные аппаратные затраты.

3.4. Понятие о минимизации логических функций

Проблема минимизации логических функций решается на основе применения законов склеивания и поглощения с последующим перебором получаемых дизъюнктивных форм и выбором из них оптимальной (минимальной). Существует большое количество методов минимизации ЛФ. Все они отличаются друг от друга спецификой применения операций склеивания и поглощения, а также различными спосо-

Табл. 3.4. Таблица истинности функции $y = f(x_1, x_2, x_3)$

x_1	x_2	x_3	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

бами сокращения переборов. Среди аналитических методов наиболее известным является метод Квайна – Мак-Класки, среди табличных – метод с применением диаграмм Вейча. Графические методы минимизации отличаются большей наглядностью и меньшей трудоемкостью. Однако их применение эффективно при малом числе переменных $n \leq 5$.

Рассмотрим последовательность действий минимизации ЛФ на примере.

Найти минимальную дизъюнктивную форму функции, заданной таблицей истинности (табл. 3.4).

Эта функция интересна тем, что имеет несколько минимальных форм. По данным таблицы запишем аналитическое выражение

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3$$

Штриховыми линиями в этом выражении отмечены пары конъюнкций, к которым можно применить операцию склеивания типа $Fx \vee F\bar{x} = F$. Особенно это видно при использовании карт Карно (диаграммы Вейча), в которой «склеиваемые» конъюнкции находятся по соседству друг с другом. Диаграмма Вейча просто по-другому интерпретирует таблицу истинности (рис. 3.3).

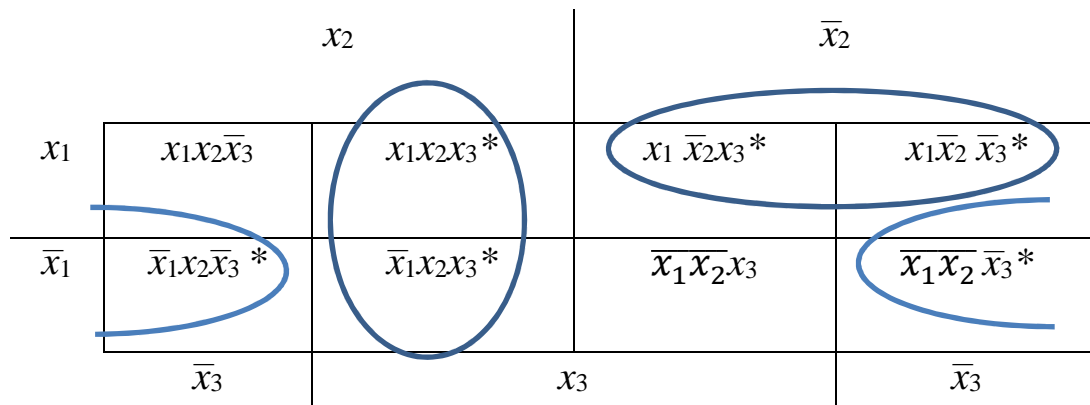


Рис. 3.3. Диаграмма Вейча функции y

После выделения конъюнкций аргументов, при которых функция принимает единичное значение (они отмечены звездочкой), видно, какие конъюнкции могут образовывать пары для склеивания.

В результате применения операций склеивания и поглощения можно получить другое аналитическое выражение

$$y = \bar{x}_1x_2 \vee x_2x_3 \vee x_1x_3 \vee x_1\bar{x}_2 \vee \bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_3,$$

в котором отсутствуют возможности дальнейших склеиваний и поглощений. Однако последнее выражение является избыточным, так как отдельные конъюнкции могут быть «лишними», т.е. их «составные части» могут включаться в другие конъюнкции.

Минимальная дизъюнктивная нормальная форма (МДНФ) может быть легко получена из диаграммы Вейча с использованием приемов минимизации на основе контуров склеивания. Контур склеивания составляется из смежных минтермов в количестве $n = 2, 4, 8, \dots$. Число n выбирается максимально возможным. В данном случае получается три контура склеивания, один из которых замыкается по внешнему контуру диаграммы. МДНФ принимает вид

$$y = \bar{x}_1 \bar{x}_3 + x_2 x_3 + x_1 \bar{x}_2.$$

Минимизация «вручную» возможна только для функций, зависящих от 4 – 5 переменных, так как трудоемкость переборных методов растет в квадратичной зависимости от числа переменных. Применение мощных ЭВМ для этих целей позволяет расширить границы до $n = 12 – 15$. Если при этом учесть, что функции могут быть частично определены (значения функций на некоторых наборах переменных можно определять произвольно), а также, что иногда приходится решать задачи совместной минимизации систем ЛФ, то минимизация ЛФ становится сложной инженерной, практической и научной проблемой.

3.5. Техническая интерпретация логических функций

По логическим выражениям проектируются схемы ЭВМ. При этом следует придерживаться последовательности действий.

1. Словесное описание работы схемы.
 2. Формализация словесного описания.
 3. Запись функций в дизъюнктивной (конъюнктивной) совершенной нормальной форме по таблицам истинности.
 4. Минимизация логических зависимостей с целью их упрощения.
 5. Представление полученных выражений в выбранном логически полном базисе элементарных функций.
 6. Построение схемы устройства.
 7. Проверка работоспособности полученной схемы.
- Покажем взаимосвязь перечисленных этапов на примере.

Пример

Спроектировать схему, фиксирующую появление «неправильной» тетрады в двоично-десятичном представлении чисел.

1. Каждая тетрада двоично-десятичного представления числа содержит десятичные цифры 0 – 9, что соответствует двоичным числам 0000 – 1001. Значения тетрады, соответствующие двоичным числам 1010 – 1111 (шестнадцатеричные цифры A – F), не должны появляться при представлении десятичных чисел.

2. Составим табл. 3.5 истинности функции, которая принимает значения, равные единице, при появлении «неправильных» тетрад. Разряды тетрады обозначим переменными x, y, z, u .

Исходная совершенная дизъюнктивная нормальная форма записывается

$$F = x\bar{y}z\bar{u} \vee x\bar{y}zu \vee xy\bar{z}\bar{u} \vee xy\bar{z}u \vee xuz\bar{y} \vee xuz\bar{y}$$

4. Эта форма функции допускает упрощение, что видно по диаграмме Вейча (рис. 3.4). Этот же результат может быть получен аналитически.

		y		\bar{y}		
x		1	1	0	0	\bar{z}
		1	1	1	1	z
\bar{x}		0	0	0	0	
		0	0	0	0	\bar{z}
		\bar{u}		u		\bar{u}

Рис. 3.4. Диаграмма Вейча для функции F

Табл. 3.5. Таблица истинности функции F

x	y	z	u	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

5. Минимальная форма функции F в логически полном базисе $\{\&, \vee\}$ будет иметь вид

$$F = xy \vee xz = x(y \vee z).$$

Для представления этой же схемы в другом полном базисе, например $\{\&\}$, воспользуемся правилом де Моргана:

$$F = xy \vee xz = \overline{\overline{xy} \cdot \overline{xz}} = \overline{\overline{xy} \cdot \overline{xz}}.$$

6. По полученным зависимостям можно построить схемы фиксации «неправильных» тетрад (рис. 3.5).

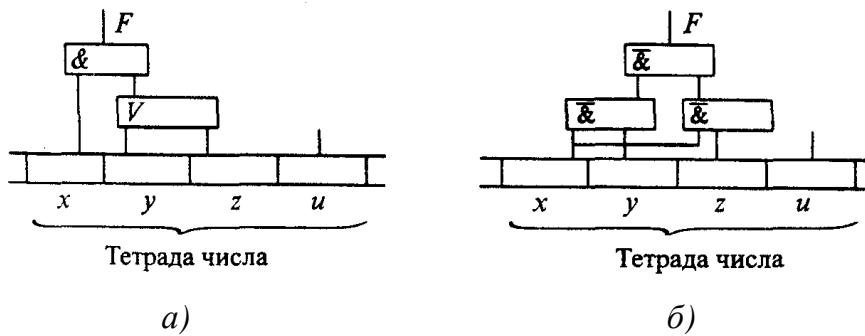
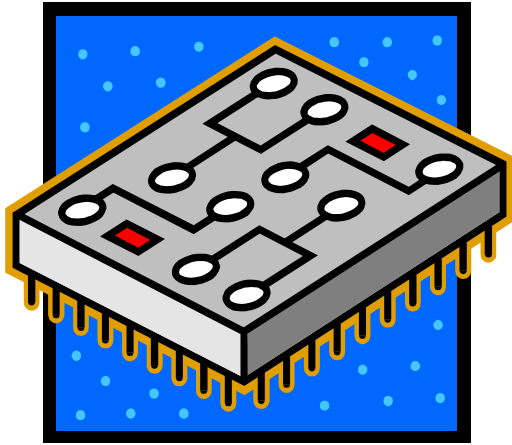


Рис. 3.5. Схемы фиксации «неправильных» тетрад
(а – схема в базисе $(\&, \vee)$, б – схема в базисе $(\&)$)

Вопросы к компьютерному тестированию

1. Дать определение понятию логическое высказывание.
2. Является ли логическим высказыванием предложение: «Основоположник дисциплины Дж. Буль – английский математик прошлого столетия»?
3. Какая из логических связок обозначается знаком «черта над высказыванием»?
4. Какая из логических связок обозначается знаком V?
5. Какая из логических связок обозначается перевернутым знаком V?
6. Какая из логических связок обозначается знаком «стрелка вправо»?
7. Какая из логических связок обозначается знаком «двойная стрелка»?
8. Как можно выразить импликацию через простейшие логические операции?
9. Как можно выразить эквиваленцию через простейшие логические операции?
10. Привести порядок выполнения логических операций при действиях со скобками.
11. Какой зависимостью выражается количество всевозможных функций N от n аргументов?

12. Как называют наборы переменных, записанные в виде конъюнкций, при которых функция принимает значение единицы?
13. Как называют наборы переменных, записанные в виде дизъюнкций, при которых функция принимает нулевое значение?
14. Как сокращенно называют набор минтермов, связанных знаками логического сложения?
15. Как сокращенно называют набор макстермов, связанных знаками логического умножения?
16. Как сокращенно называют набор минтермов, содержащих все без исключения переменные и связанных знаками логического сложения?
17. Как сокращенно называют набор макстермов, содержащих все без исключения переменные и связанных знаками логического умножения?
18. Запишите значение дизъюнкции прямого и инверсного значений переменной x .
19. Запишите значение конъюнкции прямого и инверсного значений переменной x .
20. Запишите значение дизъюнкции единицы и прямого значений переменной x .
21. Какой из законов алгебры логики иллюстрируется зависимостью $x_1 \cdot x_2 = x_2 \cdot x_1$?
22. Какой из законов алгебры логики иллюстрируется зависимостью $(x_1 \cdot x_2) \cdot x_3 = (x_1 \cdot x_3) \cdot x_2 = x_1 \cdot (x_2 \cdot x_3)$?
23. Какой из законов алгебры логики иллюстрируется зависимостью $x_1 \cdot (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3$?
24. Какой из законов алгебры логики иллюстрируется зависимостью $x_1 \vee x_1 \cdot x_2 = x_1$?
25. Какой из законов алгебры логики иллюстрируется зависимостью $x_1 x_2 \vee x_1 \bar{x}_2 = x_1$?
26. Какой из законов алгебры логики иллюстрируется зависимостью $x \vee \bar{x} F = x \vee F$?
27. Какой из законов алгебры логики иллюстрируется зависимостью $\overline{x_1 \vee x_2} = \bar{x}_1 \cdot \bar{x}_2$?
28. Привести первое определение правила Де Моргана.
29. Какие из законов алгебры логики положены в основу минимизации логических функций методом Квайна?
30. Привести второе определение правила Де Моргана.



Глава 4

ЭЛЕМЕНТНАЯ БАЗА ЭВМ

Рассматриваемые вопросы:

- 4.1. Классификация элементов и узлов ЭВМ.
- 4.2. Комбинационные схемы.
- 4.3. Последовательностные схемы.
- 4.4. Развитие элементной базы ЭВМ.

4.1. Классификация элементов и узлов ЭВМ

Технические и программные средства ЭВМ взаимосвязаны и объединяются в одну структуру.

Структура – совокупность элементов и их связей. Различают структуры технических, программных и аппаратурно-программных средств.

При рассмотрении структуры любой ЭВМ можно выделить следующие структурные единицы: *устройства, узлы, блоки и элементы*.

Элементы предназначены для обработки единичных электрических сигналов, соответствующих битам информации (например, ячейка памяти).

Узлы обеспечивают одновременную обработку группы сигналов – информационных слов (пример – регистры).

Блоки реализуют функционально обособленную часть машинных операций (блок выборки команд, блок записи-чтения и др.).

Устройства предназначаются для выполнения отдельных машинных операций и их последовательностей (запоминающее устройство).

Элементы ЭВМ можно *классифицировать* по различным признакам. Наиболее часто такими признаками являются:

- тип сигналов;
- назначение элементов;
- технология их изготовления и т.д.

В ЭВМ широко применяют два способа физического представления сигналов: *импульсный* и *потенциальный*. При *импульсном* способе представления сигналов единичному значению некоторой двоичной переменной ставится в соответствие наличие импульса (тока или напряжения), нулевому значению – отсутствие импульса (рис. 4.1, а). Длительность импульсного сигнала не превышает один такт синхроимпульсов. При *потенциальном* или статическом представлении сигналов единичное значение двоичной переменной отображается высоким уровнем напряжения, а нулевое значение – низким уровнем (рис. 4.1, б).

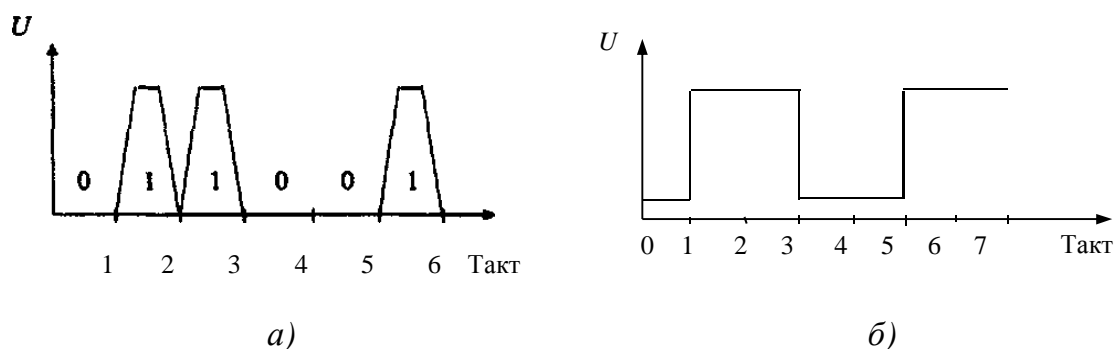


Рис. 4.1. Представление сигналов

Независимо от вида сигналов различают *последовательный* и *параллельный* коды передачи и представления информации в ЭВМ.

При *последовательном коде* представления данных используются одиночные шины или линии передачи, в которых сигналы, соответствующие отдельным разрядам данных, разнесены во времени. Обработка такой информации производится последовательно разряд за разрядом. Время обработки определяется числом обрабатываемых сигналов (разрядов).

Параллельный код отображения и передачи информации предполагает параллельную и одновременную фиксацию всех разрядов данных на различных шинах, т.е. параллельный код данных развернут в пространстве. Это дает возможность ускорить обработку во времени, но затраты на аппаратурные средства при этом возрастают пропорционально числу обрабатываемых разрядов.

Во всех вычислительных машинах используются и **параллельно-последовательные коды** представления информации. При этом информация отображается частями. Части поступают на обработку последовательно, а каждая часть данных представляется параллельным кодом.

По своему назначению элементы делятся на *формирующие, логические и запоминающие*.

К **формирующим элементам** относятся различные формирователи, усилители, усилители-формирователи и т.п. Данные элементы служат для выработки определенных электрических сигналов, восстановления их параметров (амплитуды, полярности, мощности, длительности).

Простейшие **логические элементы** преобразуют входные сигналы в соответствии с элементарными логическими функциями. В свою очередь, полученные сигналы могут формировать следующий уровень сигналов и т. д. Сложные преобразования в соответствии с требованиями логическими зависимостями могут приводить к построению многоуровневых схем. Каждая такая схема представляет собой композицию простейших логических схем.

Запоминающим элементом называется элемент, который способен принимать и хранить код двоичной цифры (единицы или нуля). Элементы памяти могут запоминать и сохранять исходные значения некоторых величин, промежуточные значения обработки и окончательные результаты вычислений.

4.2. Комбинационные схемы

Обработка входной информации X в выходную Y в любых схемах ЭВМ обеспечивается преобразователями, или цифровыми автоматами, двух видов: **комбинационными схемами (КС)** и **последовательными схемами (ПС)** – схемами с памятью.

Комбинационные схемы – это схемы, у которых выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в любой момент дискретного времени однозначно определяются совокупностью входных сигналов $X = (x_1, x_2, \dots, x_n)$, поступающих в тот же момент времени t . Реализуемый в КС способ обработки информации называется комбинационным потому, что резуль-

тат обработки зависит только от комбинации входных сигналов и формируется сразу при поступлении входных сигналов. Поэтому одно из достоинств комбинационных схем – высокое быстродействие. Преобразование информации однозначно описывается логическими функциями вида $Y = f(X)$.

Логические функции и соответствующие им комбинационные схемы подразделяют на **регулярные и нерегулярные структуры**. *Регулярные* структуры предполагают построение схемы таким образом, что каждый из ее выходов выполнен по аналогии с предыдущими. В *нерегулярных* структурах такая аналогия отсутствует.

В практике проектирования ЭВМ накоплен огромный опыт по синтезу различных схем [8]. Многие регулярные структуры положены в основу построения отдельных ИС малой и средней степени интеграции или отдельных функциональных частей БИС и СБИС. Из регулярных комбинационных схем наиболее распространены дешифраторы, шифраторы, схемы сравнения, комбинационные сумматоры, коммутаторы и др.

Рассмотрим принципы построения подобных регулярных структур.

Дешифраторы (ДШ) – это комбинационные схемы с n входами и $m = 2^n$ выходами. Единичный сигнал, формирующийся на одном из m выходов, однозначно соответствует комбинации входных сигналов. Например, разработка структуры ДШ для $n = 3$ позволяет получить таблицу истинности (табл. 4.1) и соответствующие логические зависимости.

Табл. 4.1. Таблица истинности дешифратора

Входы			Выходы					
X_1	X_2	X_3	Y_0	Y_1	...	Y_5	...	Y_7
0	0	0	1	0	...	0	...	0
0	0	1	0	1	...	0	...	0
0	1	0	0	0	...	0	...	0
0	1	1	0	0	...	0	...	0
1	0	0	0	0	...	0	...	0
1	0	1	0	0	...	1	...	0
1	1	0	0	0	...	0	...	0
1	1	1	0	0	...	0	...	1

Дешифраторы широко используются в ЭВМ для выбора информации по определенному адресу, расшифровки кода операции и др. Логические зависимости дешифратора:

$$\begin{aligned}
 Y_0 &= \bar{x}_1 \bar{x}_2 \bar{x}_3, & Y_4 &= x_1 \bar{x}_2 \bar{x}_3, \\
 Y_1 &= \bar{x}_1 \bar{x}_2 x_3, & Y_5 &= x_1 \bar{x}_2 x_3, \\
 Y_2 &= \bar{x}_1 x_2 \bar{x}_3, & Y_6 &= x_1 x_2 \bar{x}_3, \\
 Y_3 &= \bar{x}_1 x_2 x_3, & Y_7 &= x_1 x_2 x_3.
 \end{aligned}$$

На рис. 4.2 представлены структурная схема ДШ (а), построенная в базисе (И, НЕ), и условное ее обозначение (б) на принципиальных электрических схемах ЭВМ.

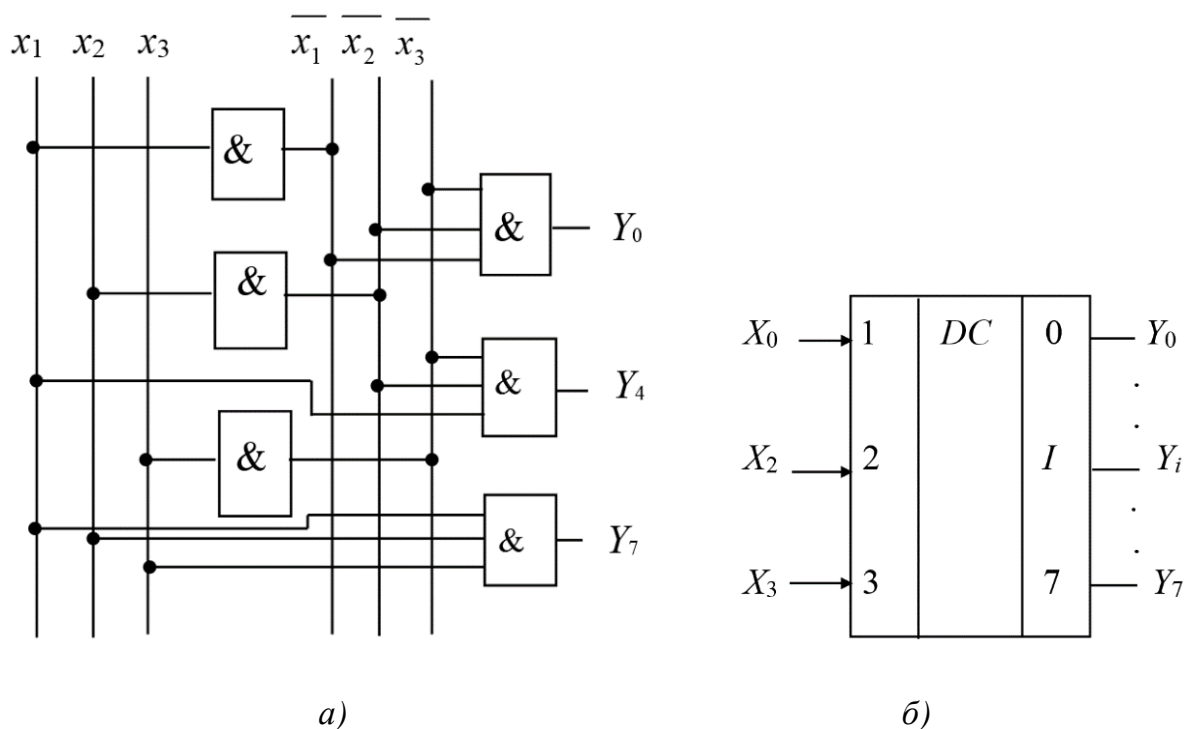


Рис. 4.2. Дешифратор

Шифратор (ШР) решает задачу, обратную схемам ДШ, т. е. по номеру входного сигнала формирует однозначную комбинацию выходных сигналов. Пример построения ШР иллюстрируется таблицей истинности (табл. 4.2), соответствующими логическими зависимостями и схемами на рис. 4.3.

Табл. 4.2. Таблица истинности шифратора

Входы							Выходы		
X_1	X_2	X_3	X_4	X_5	X_6	X_7	Y_0	Y_1	Y_2
1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	1	1	1	1

Логические зависимости:

$$Y_2 = x_1 \vee x_3 \vee x_5 \vee x_7, Y_1 = x_2 \vee x_3 \vee x_6 \vee x_7, Y_0 = x_4 \vee x_5 \vee x_6 \vee x_7.$$

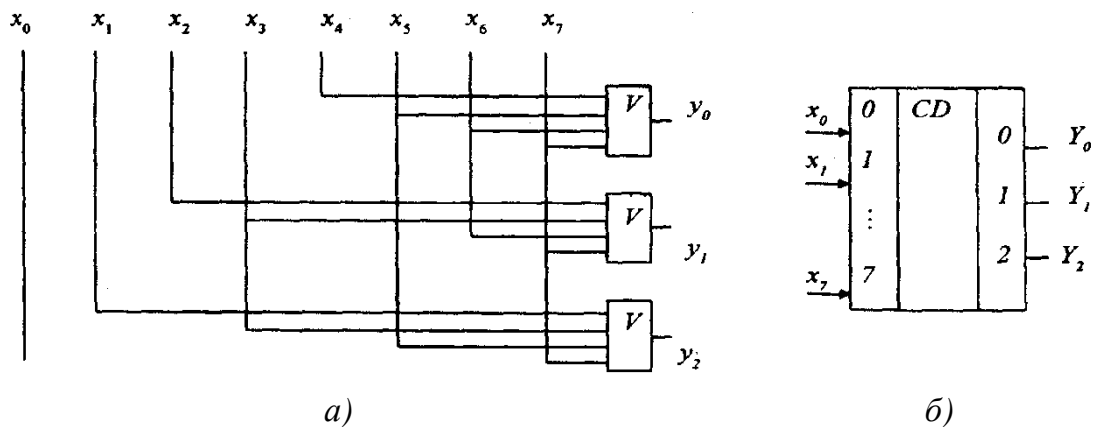


Рис. 4.3. Структурная схема шифратора (а) и обозначение дешифратора на принципиальных электрических схемах (б)

Обратим внимание, что табл. 4.2 и 4.1 во многом похожи, входы и выходы в них поменялись местами. Состояния входов табл. 4.2 содержат только по одному единичному элементу. Другие произвольные комбинации входов недопустимы.

Мультиплексор – операционный узел, состоящий из устройств, в которых обрабатывается и хранится информация, а именно счетчиков, регистров, шифраторов, дешифраторов, сумматоров, схем сравнения, блоков памяти и т.п. Используя набор перечисленных устройств, можно производить обработку поступающей информации, результатом которой является выходная информация.

Мультиплексором называется комбинационная схема, имеющая $m + 2m$ входов и один выход, где m – количество адресных входов,

а $2m$ – количество информационных входов мультиплексора. Адреса представляются в двоичном коде, и им присваивается номер j . Каждому адресу с номером j соответствует свой информационный вход A_j , сигнал с которого при данном адресе проходит на выход. Основное назначение мультиплексора – коммутация $2m$ входных сигналов на один выход.

Условное описание мультиплексора задаётся формулой $2m \times 1$. Примерами являются мультиплексоры: 2×1 , 4×1 , 8×1 , 16×1 .

В состав мультиплексора обычно включают двоичный дешифратор, как показано на рис. 4.4. Это позволяет управлять переключением информационных входов при помощи двоичных кодов, подаваемых на управляющие входы. Количество информационных входов в таких схемах выбирают кратным степени числа два.

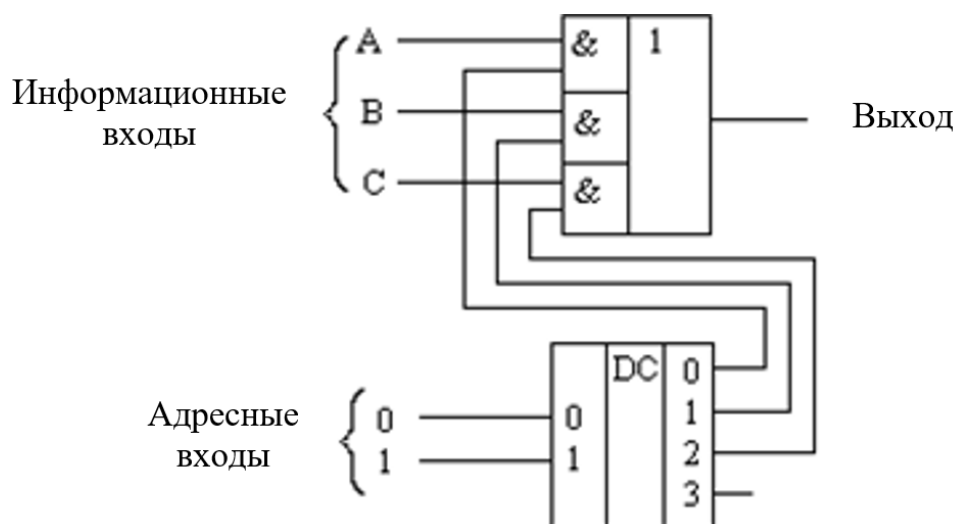


Рис. 4.4. Схема подключения мультиплексора с дешифратором

Условное графическое обозначение четырёхвходового мультиплексора с двоичным управлением приведено на рис. 4.5. Входы A0 и A1 являются управляющими входами мультиплексора, определяющими адрес входного сигнала, который будет соединён с выходным выводом мультиплексора Y. Сами входные сигналы обозначены как X0, X1, X2 и X3.

Демультимплексор – устройство, позволяющее подключать один вход к нескольким выходам. Его можно построить на основе точно таких же схем ло-

гического "И", как и при построении мультиплексора. Существенное отличие от мультиплексора – возможность объединения нескольких входов в один без дополнительных схем. Однако для увеличения нагрузочной способности микросхемы на входе демультимплексора для усиления входного сигнала ставят инвертор.

Схема демультимплексора приведена на рис. 4.6. В этой схеме для выбора конкретного выхода демультимплексора, как и в мультиплексоре, используется двоичный дешифратор. Условно-графическое изображение демультимплексора приведено на рис. 4.7.

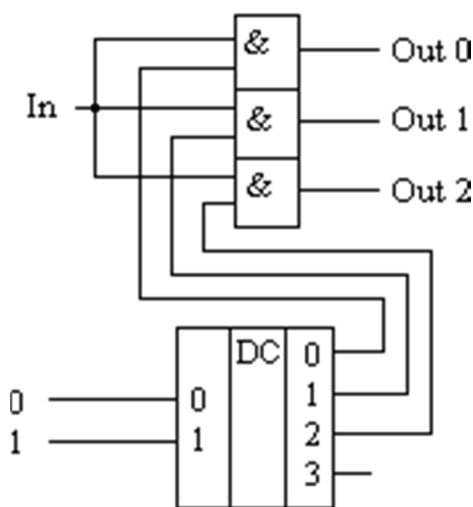


Рис. 4.6. Схема демультимплексора

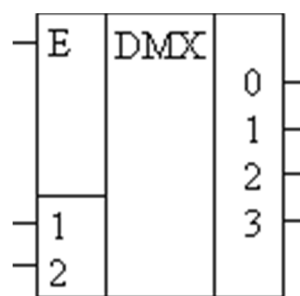


Рис. 4.7. Демультимплексор

В изображении демультимплексора вход обозначен как E , а выходы пронумерованы.

Схемы сравнения, или компараторы, обычно строятся как поразрядные.

Они широко используются и автономно, и в составе более сложных схем, например при построении сумматоров. Таблица истинности (табл. 4.3) отражает логику работы i -го разряда компаратора при сравнении двух векторов A и B . На рис. 4.8

показана структурная схема компаратора (a) и обозначение его на схемах (b).

Табл. 4.3. Таблица истинности компаратора

Входы		Выходы
A_i	B_i	Y_i
0	0	1
0	1	0
1	0	0
1	1	1

Логическая зависимость компаратора

$$Y = \overline{a_i b_i} \vee a_i b_i = \overline{\overline{a_i b_i}} = \overline{a_i \oplus b_i}. \quad (4.1)$$

На рис. 4.8 помимо выхода Y_2 , фиксирующего равенство значений разрядов, показаны выходы Y_1 и Y_3 , соответствующие сигналам "больше" и "меньше".

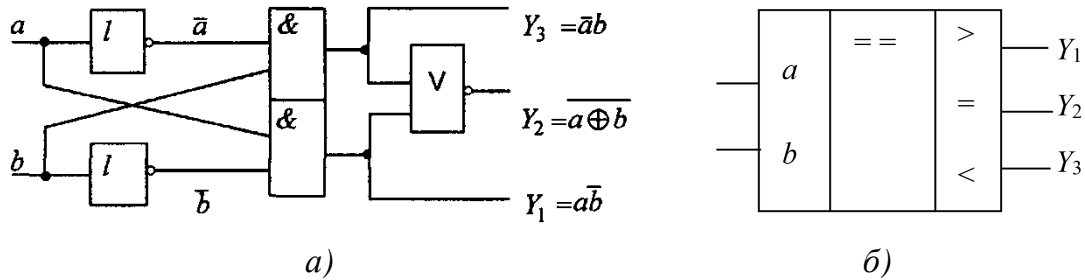


Рис. 4.8. Структурная схема компаратора (а) и обозначение компаратора на принципиальных электрических схемах (б)

Комбинационный сумматор. Принципы построения и работы сумматора вытекают из правил сложения двоичных цифр. Схема сумматора также является регулярной и широко используется в ЭВМ. При сложении одноразрядных двоичных цифр можно выявить закономерности в построении и многоразрядных сумматоров.

Табл. 4.4. Таблица истинности комбинационного полусумматора

Входы		Выходы	
a_i	b_i	S_i	p_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

висимостями (4.2), где S_i – функция одноразрядной суммы и p_i – функция формирования переноса. Перенос формируется в том случае, когда $a_1 = 1$ и $b_1 = 1$.

Логические зависимости:

$$S_i = \overline{a_i b_i} \vee a_i \bar{b}_i = a_i \oplus b_i; \quad P_i = a_i b_i. \quad (4.2)$$

Зависимости соответствуют логике работы самого младшего разряда любого сумматора. Структурная схема одноразрядного сумма-

тора (полусумматора) (*a*) и его обозначение на принципиальных схемах (*б*) представлены на рис. 4.9.

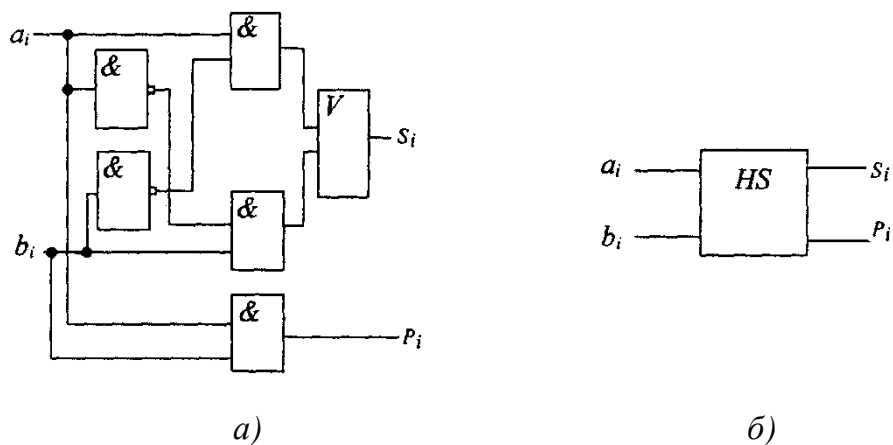


Рис. 4.9. Одноразрядный сумматор

Логические зависимости полусумматора S_i и компаратора очень похожи, так как они инверсны по отношению друг к другу.

Уравнения, положенные в основу одноразрядного сумматора, используются и при построении многоразрядных сумматоров. Логика работы каждого разряда сумматора описывается табл. 4.5, которую можно считать его таблицей истинности.

Таблица истинности сумматора, учитывающего сигналы переноса, отличается от таблицы полусумматора (см. табл. 4.4) дополнительным входом (p -переносом) из предыдущих разрядов.

Табл. 4.5. Правила сложения двоичных цифр

Значения двоичных чисел A и B			Сумма	Перенос в следующий разряд
$a >$	b_i	P_{i-1}		
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Исходные логические зависимости, формируемые по табл. 4.5, имеют следующие совершенные ДНФ:

$$S_i = \overline{a}b\overline{p} \vee \overline{a}b\overline{p} \vee \overline{a}b\overline{p} \vee abp;$$

$$P_i = \overline{a}bp \vee \overline{a}b\overline{p} \vee ab\overline{p} \vee abp.$$

Преобразование этих выражений приводит к следующим зависимостям:

$$S_i = p(\overline{ab} \vee ab) \vee b(\overline{ap} \vee ap) \vee a(\overline{bp} \vee bp) = p(\overline{a \oplus b}) \vee b(\overline{a \oplus p}) \vee a(\overline{b \oplus p});$$

$$P_i = ab \vee ap \vee bp. \quad (4.3)$$

В приведенных выражениях индексы у переменных в правых частях уравнений опущены.

Из анализа логических зависимостей видно, что схема i -го разряда сумматора требует включения в свой состав трех схем сравнения для формирования разрядной суммы и шести схем совпадения. Структурная схема сумматора (а) и его обозначение на принципиальных схемах (б) на электрических схемах показана на рис. 4.10.

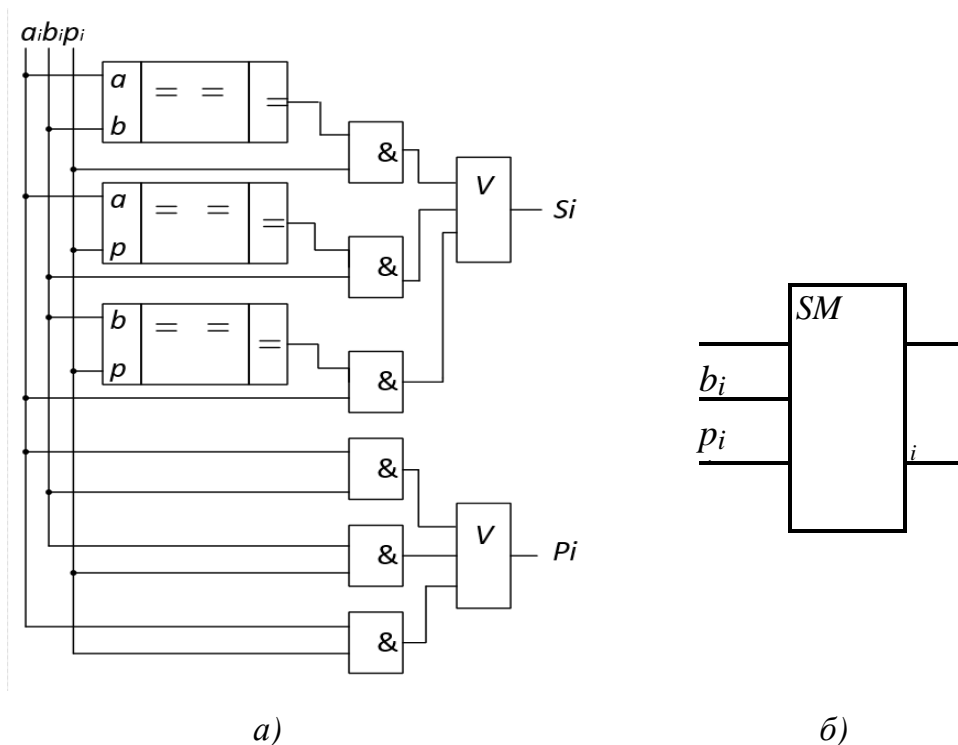


Рис. 4.10. Комбинационный сумматор

Структурная схема многоразрядного комбинационного сумматора на электрических схемах показана на рис. 4.11.

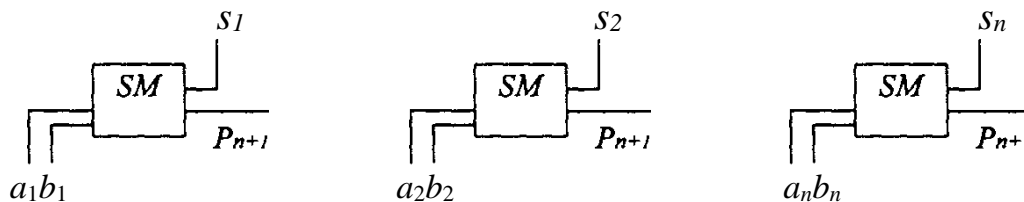


Рис. 4.11. Многоразрядный комбинационный сумматор

4.3. Последовательностные схемы

Наличие памяти в таких схемах позволяет запоминать промежуточные состояния обработки и учитывать их значения в дальнейших преобразованиях. Выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в схемах данного типа формируются не только по совокупности входных сигналов $X = (x_1, x_2, \dots, x_n)$, но и по совокупности состояний схем памяти $Q = (q_1, q_2, \dots, q_k)$. При этом различают дискретные текущий t и последующий $(t + 1)$ моменты времени (рис. 4.12).

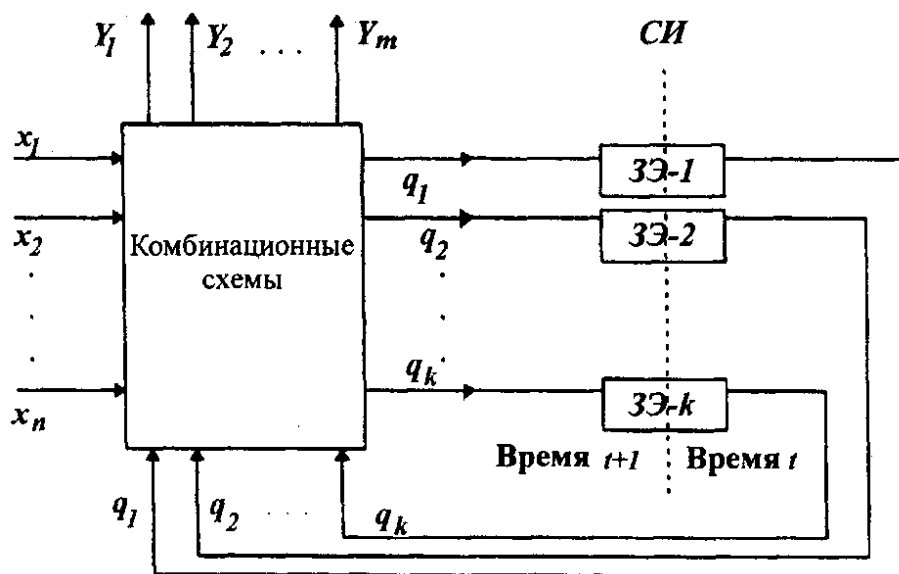


Рис. 4.12. Обобщенная структура схемы с памятью

Передача значения Q между моментами времени t и $(t + 1)$ осуществляется с применением двухступенчатой памяти и синхронизирующих импульсов (СИ).

В качестве простейшего запоминающего элемента (ЗЭ) в современных ЭВМ используют триггеры.

Триггер – устройство, которое может находиться в одном из двух устойчивых состояний и сохранять его теоретически бесконечно долго. Любой триггер является схемой с памятью или автоматом. Переключение триггера происходит по входному сигналу извне. По способу синхронизации различают синхронные и асинхронные триггерные схемы. По структурному построению – однотактные (триггеры за-

щелки), двухтактные и триггеры с динамическим управлением. По способу реакции на помехи – прозрачные и непрозрачные. Непрозрачные, в свою очередь, делятся на проницаемые и непроницаемые.

Наиболее широкое распространение в схемах ЭВМ нашел *RS*-триггер. Одиночные триггеры этого типа часто используются в различных блоках управления. В асинхронных *RS*-триггерах имеется один существенный недостаток, обусловленный логикой их построения, т.е. в них сигналы *R* и *S* должны быть разнесены во времени. Дополнение этого триггера комбинационными схемами синхронизации на входе и выходе позволяет получить триггеры с более сложной логикой работы: синхронные *RS*-триггеры, *T*-, *JK*-, *D*-триггеры и целый ряд комбинированных *RST*-, *JKRS*-, *DRS*-триггеров.

Прописные буквы в названиях триггеров обозначают:

- *R* (*Reset* – сброс) – вход установки триггера в нулевое состояние $Q = 0$;
- *S* (*Set* – установка) – вход установки триггера в единичное состояние $Q = 1$;
- *T* (*Toggle* – релаксатор) – счетный вход триггера;
- *J* (*Jerk* – внезапное включение) – вход установки *JK*-триггера в единичное состояние $Q = 1$;
- *K* (*Kill* – внезапное выключение) – $Q = 0$;
- *D* (*Delay* – задержка) – вход установки триггера в единичное или нулевое состояние на время, равное одному такту;
- *C* (*Clock* – часы) – вход синхронизирующих тактовых импульсов.

Каждый из типов триггеров можно описать таблицей работы (таблицей истинности), в которой указывается, как различные значения на входах триггера влияют на его состояние. Состояние триггера обозначают буквой *q*. Индекс *t* возле буквы означает состояние до подачи или после подачи сигнала. Ниже приведены таблицы для перечисленных триггеров в асинхронном режиме (без входа *C*).

Если триггер синхронный, то существует также дополнительный вход синхронизации. При записи информации в триггер на него необходимо подать единичный сигнал.

RS-триггер

Зависимость выходных сигналов RS-триггера от входных иллюстрируется табл. 4.6.

Табл. 4.6. Таблица истинности RS-триггера

Входные сигналы		Состояние q^t		Режим
		0	1	
R	S	Состояние q^{t+1}		
0	0	0	1	Хранение
0	1	1	1	Установка 1
1	0	0	0	Установка 0
1	1	*	*	Неопределенность

На рис. 4.13 показаны схемы синхронного однокаскадного (а) и двухкаскадного (б) RS-триггеров.

Двухкаскадная схема RS-триггера нашла наиболее широкое применение для построения n -разрядных схем запоминания – всевозможных регистровых схем. Штриховыми линиями на схеме указаны дополнительные точки подключения сигналов установки и сброса.

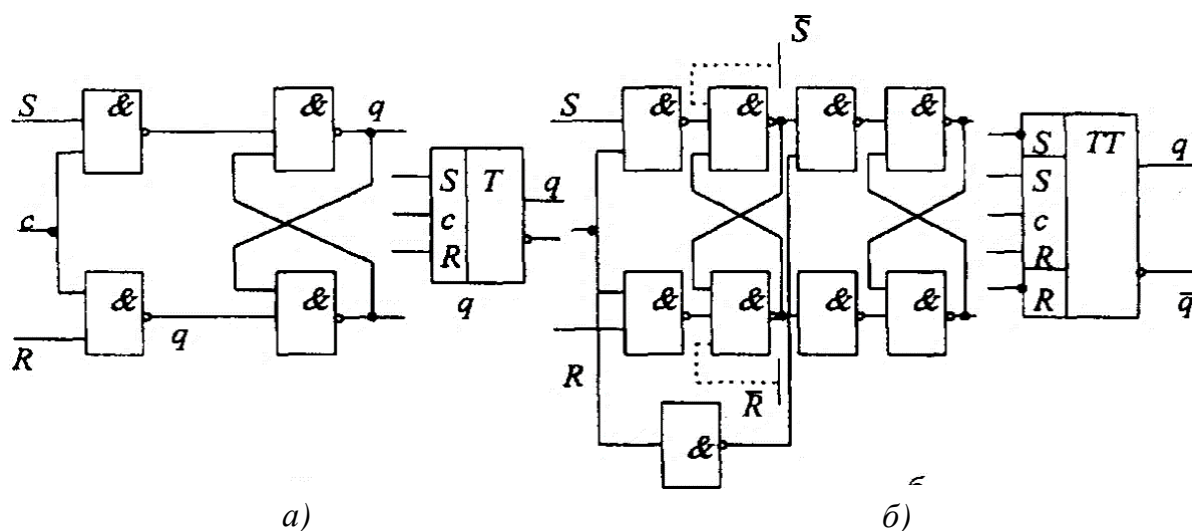


Рис. 4.13. Электрическая и функциональная схемы синхронных RS-триггеров:
а, б – варианты

На рис. 4.14 приведена схема T -триггера, или иначе – триггера со счетным входом. При значении $T = 0$ триггер сохраняет свое ранее установленное состояние – режим хранения состояния, при $T = 1$ триггер переходит в противоположное состояние. Таблица переходов (табл. 4.7) и диаграмма работы (рис. 4.14, в) отражают динамику работы этого элемента.

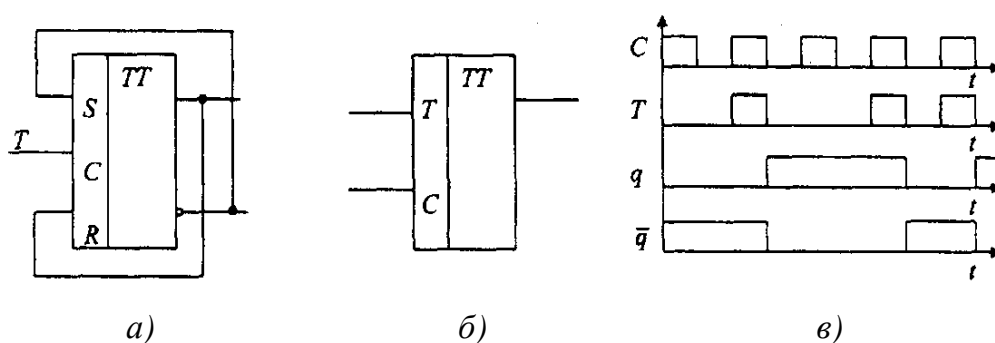


Рис. 4.14. Схема, обозначение и диаграммы работы T -триггера

Табл. 4.7. Таблица переходов T -триггера

Входные сигналы	Состояние q_t		Режим
X_t	0	1	
0	0	1	Хранение
1	1	0	Инверсия

По таблице переходов можно получить логическую функцию, реализуемую T -триггером:

$$Q_{t+1} = \overline{q_t} \cdot x_t \vee q_t \cdot \overline{x_t} = q_t \oplus x_t.$$

Нетрудно видеть, что эта зависимость очень похожа на функцию, выведенную для одноразрядного комбинационного полусумматора. На рис. 4.14, а показано, как двухтактный RS -триггер преобразуется в T -триггер.

Наиболее сложным типом триггера является JK -триггер. Он, по существу, является объединением двухтактного RS - и T -триггеров. Этому соответствует его таблица переходов (табл. 4.8).

Табл. 4.8. Таблица переходов *JK*-триггера

Входные сигналы		Состояние q		Режим
J	K	0	1	
0	0	0	1	Хранение
0	1	0	0	Установка 0
1	0	1	1	Установка 1
1	1	1	0	Инверсия

Если первые три строки таблицы переходов полностью повторяют соответствующие строки табл. 4.6, то последняя строка с запрещенной комбинацией для *RS*-триггера соответствует режиму переключения *T*-триггера (см. табл. 4.7). Схема *JK*-триггера изображена на рис. 4.15.

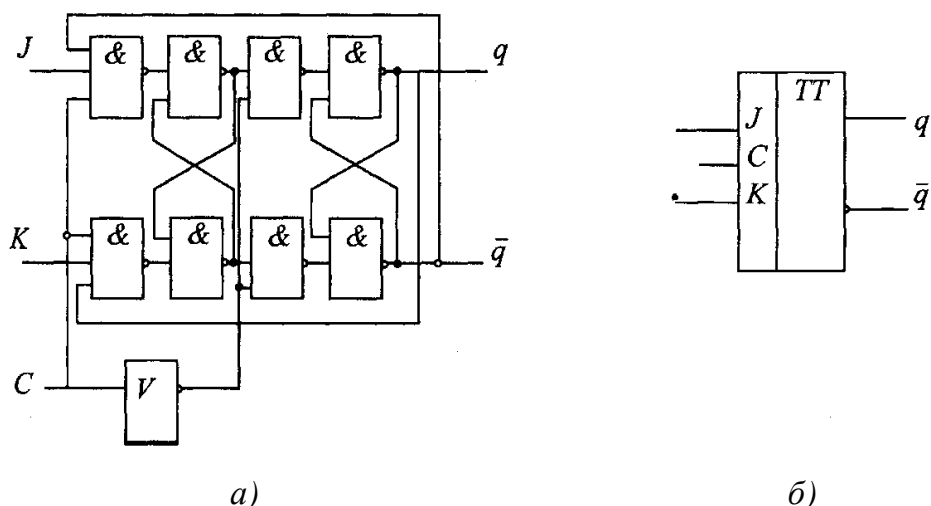


Рис. 4.15. *JK*-триггер: *а* — функциональная схема; *б* — условное обозначение

D-триггер обычно строится на основе двухтактного *RS*- или *JK*-триггера. Он предназначен для хранения состояния (1 или 0) на один период тактовых импульсов (с задержкой на один такт). Таблица его переходов отражена в табл. 4.9. На рис. 4.16, *а* и *б* представлены варианты его построения, а на рис. 4.16, *в* — его условное обозначение.

Табл. 4.9. Таблица переходов *D*-триггера

Входные сигналы D	Состояние q		Режим
	0	1	
0	0	0	Установка 0
1	1	1	Установка 1

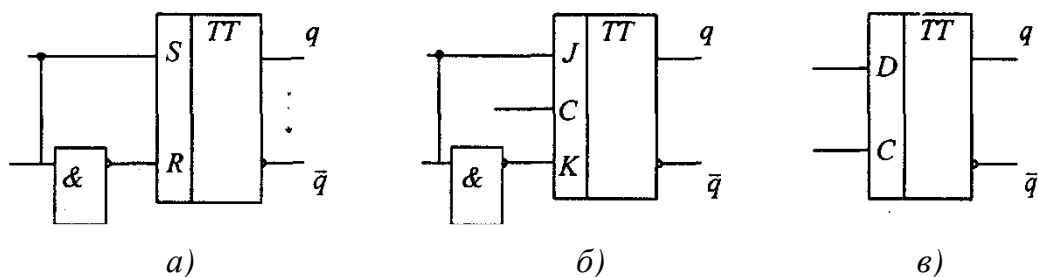


Рис. 4.16. D-триггер: *а* – функциональная схема на основе RS-триггера;
б – функциональная схема на основе JK-триггера;
в – условное обозначение

Все перечисленные элементы памяти позволяют хранить одну единицу информации – бит или одну двоичную цифру.

При построении ЭВМ широко используются функциональные схемы, обеспечивающие операции хранения и преобразования информации над группами битов (машинными словами). Такие сложные схемы называются узлами. К типовым узлам относят: регистры, счетчики, сумматоры. Все они также принадлежат к регулярным структурам, состоящим из одинаковых параллельно работающих одноразрядных схем.

Регистром называется узел, предназначенный для приема, временного хранения и выдачи машинного слова. Регистры могут также использоваться для некоторых операций преобразования данных: сдвига кода числа (слова) на определенное число разрядов влево или вправо, преобразования последовательного кода числа в параллельный и наоборот и т.д. Эти дополнительные функции регистров обеспечиваются путем усложнения схем хранения, выбора более сложных триггеров и подключения дополнительных логических схем на их входах и выходах.

Таким образом, регистры представляют собой совокупность триггеров, число которых соответствует числу разрядов в слове, и вспомогательных схем, обеспечивающих выполнение различных операций над словом.

На рис. 4.17 показана функциональная схема *n*-разрядного регистра, построенного на RS-триггерах. Информация в регистр записыва-

ется под действием сигнала "Запись". Предварительно перед установкой кода на регистр обычно на все разряды R подается сигнал сброса. На рисунке показано, что подключение к входам R дополнительных инверторов позволяет избежать этой предварительной операции. Здесь на вход каждого разряда поступает парафазный код двоичной цифры (x_i – на вход S_i и \bar{x}_i – на вход R_i), т.е. прямое и инверсное значения кода подаются в противофазе.

На вход каждого разряда поступает парафазный код двоичной цифры (x_i – на вход S_i и \bar{x}_i – на вход R_i), т.е. прямое и инверсное значения кода подаются в противофазе.

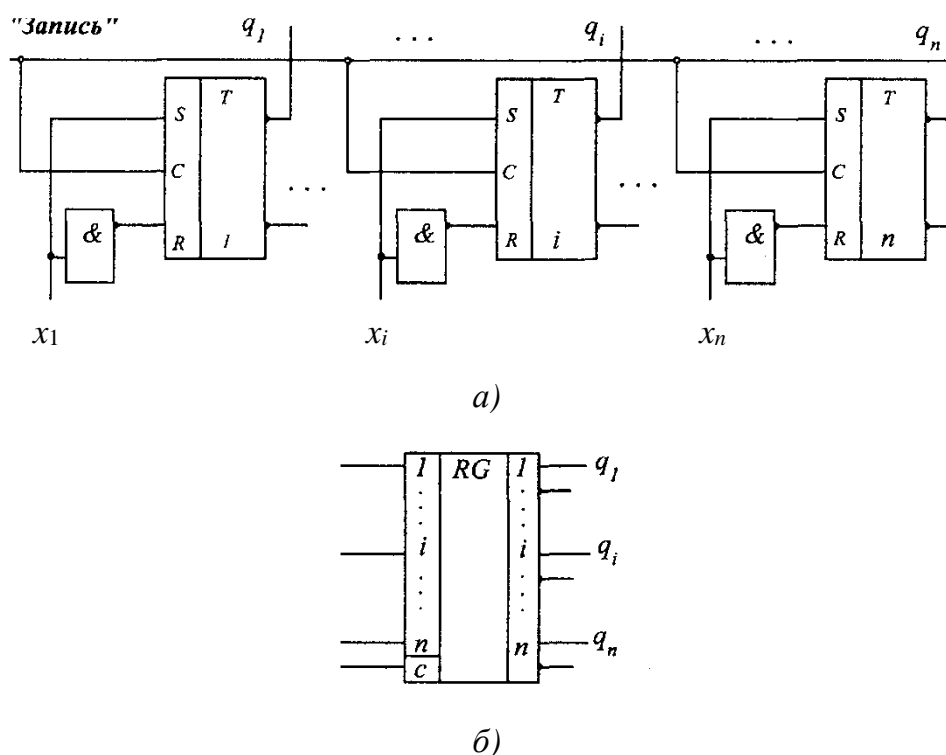


Рис. 4.17. Схема регистра на RS -триггерах: а – функциональная схема; б – условное обозначение

Счетчик – узел ЭВМ, позволяющий подсчитывать поступающие на его вход сигналы и фиксировать результаты в виде многоразрядного двоичного числа. Счетчик, состоящий из n -триггеров, дает возможность подсчитывать до $N = 2^n$ сигналов, связанных зависимостью $n = \log_2 N$.

В ЭВМ счетчики используются для подсчета импульсов, сдвигов, формирования адресов и т.д. Функционально различают суммирующие, вычитающие, реверсивные счетчики. Они также отличаются друг от друга логикой работы дополнительных логических элементов, подключаемых к триггерам.

В основу построения любого счетчика положено свойство T -триггеров изменять свое состояние при подаче очередного сигнала на счетный вход T . На рис. 4.18 показана схема трех разрядов суммирующего счетчика, построенного на T -триггерах. Логика его работы представлена в табл. 4.10.

Табл. 4.10. Таблица переходов трехразрядного счетчика

Вход x	Состояние								Режим
	000	001	010	011	100	101	110	111	
0	000	001	010	011	100	101	110	111	Хранение
1	001	010	011	100	101	110	111	000	Счет

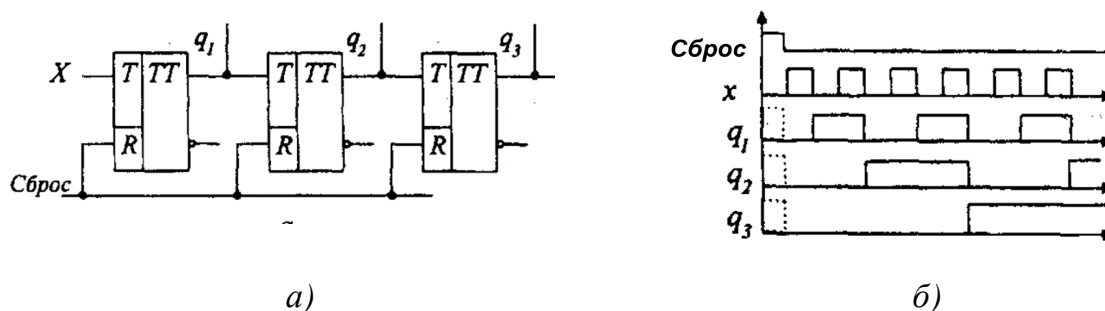


Рис. 4.18. Организация счетчика на T -триггерах: а – функциональная схема; б – временная диаграмма

Сумматор – узел ЭВМ, в котором суммируются коды чисел. Как правило, любой сумматор представляет собой комбинацию одноразрядных сумматоров. Сумматоры различают по принципам построения – накапливающего или комбинационного типа. Сумматоры накапливающего типа строят на сложных $JKRS$ -триггерах, дополняя их выходы достаточно сложными схемами формирования и распространения переносов. Процесс сложения при этом осуществляется поэтапно. Сначала на триггерах сумматора фиксируется код первого операнда, затем на

счетные коды разрядов подается код второго операнда. По зависимостям (4.3) на каждом триггере формируются одноразрядные суммы и значения переносов между разрядами. Учет возникающих переносов задерживает формирование окончательного результата суммы и может требовать дополнительных тактов сложения. Из-за этого многоразрядные схемы сумматора накапливающего типа используются достаточно редко.

Более часто для построения сумматоров используются сумматоры комбинационного типа. Логика работы такого сумматора была представлена данными табл. 4.5. Обычно у такого сумматора на входе и выходе имеются регистры для хранения и преобразования кодов операндов и результата (рис. 4.19).

Регистр Rr1 предназначается для хранения кода первого операнда, регистр Rr2 – для хранения кода второго операнда. Сумматор по сигналам из устройства управления настраивается на выполнение определенной машинной операции, соответствующей коду операции, находящемуся в коде команды. Результат выполняемой операции фиксируется в регистре Rr3. При необходимости этот результат может использоваться для продолжения вычислений. Для этого предусматривается возможность перезаписи содержимого регистра Rr3 на Rr1 в качестве значения одного из операндов при выполнении очередной операции.

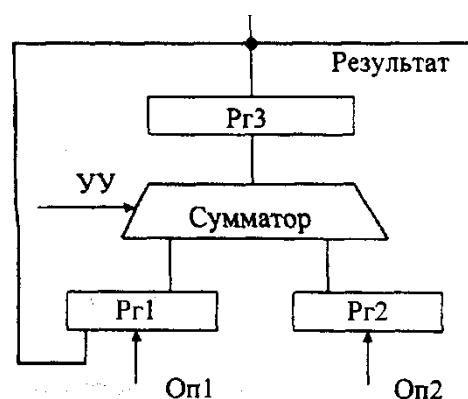


Рис. 4.19. Упрощенная схема сумматора ЭВМ

4.4. Развитие элементной базы ЭВМ

Одним из главных факторов достижения высокого быстродействия, а значит, и высокой производительности ЭВМ является построение их на новейшей элементной базе.

Все современные ЭВМ строятся на микропроцессорных наборах, основу которых составляют большие и сверхбольшие интегральные схемы. Технологический принцип разработки и производства инте-

графических схем действует уже более четверти века. Он заключается в циклическом послойном изготовлении частей электронных схем по циклу программа – рисунок – схема. По специальным программам на напыленный фоторезистивный слой наносится рисунок будущего слоя микросхемы. Затем рисунок протравливается, фиксируется, закрепляется и изолируется от новых слоев. На основе этого создается пространственная твердотельная структура.

Степень микроминиатюризации, размер кристалла ИС, производительность и стоимость технологии напрямую определяются типом литографии. До настоящего времени доминирующей оставалась оптическая литография, т.е. послойные рисунки на фоторезисторе микросхем наносились световым лучом. В настоящее время ведущие компании, производящие микросхемы, реализуют кристаллы с размерами менее 200 мм^2 – для процессоров (*Cyrix*) и 50 мм^2 – для схем памяти. Первые процессоры *Pentium* (1993 г.) производились по техпроцессу (т.е. минимальному топологическому размеру (толщине линий) $0,8 \text{ мкм}$, затем по $0,6 \text{ мкм}$. В 1995 г. впервые для процессоров 6-го поколения был применен техпроцесс $0,35 \text{ мкм}$. В 1997 г. он сменился на $0,25 \text{ мкм}$, а в 1999 – на $0,18 \text{ мкм}$. Современные процессоры выполняются по технологии $0,13$ и $0,09 \text{ мкм}$.

Техпроцесс определяет количество транзисторов в кристалле. Закон Мура (1975 г.) гласит, что число интегральных транзисторов на кристаллах микросхем (прежде всего микропроцессоров) будет удваиваться каждые полтора-два года. Например, *Pentium M* содержит 77 млн транзисторов. Двухъядерный процессор *Core 2*, использующий 45-нм техпроцесс *Intel*, содержит уже 410 млн транзисторов. Для сравнения можно привести такой пример. Толщина человеческого волоса составляет примерно 100 мкм . Значит, при данном разрешении на толщине волоса могут вычерчиваться более семисот линий. При таких высоких технологиях возникает целый ряд проблем. Микроскопическая толщина линий, сравнимая с диаметром молекул, требует высокой чистоты используемых и напыляемых материалов, применения вакуумных установок и снижения рабочих температур. Действительно, достаточно попадания мельчайшей пылинки при изготовлении микросхемы,

как она становится бракованной. Поэтому новые заводы по производству микросхем имеют уникальное оборудование, размещаемое в чистых помещениях класса 1, микросхемы в которых транспортируются от оборудования к оборудованию в замкнутых сверхчистых мини-атмосферах класса 1000. Мини-атмосфера создается, например, сверхчистым азотом или другим инертным газом при давлении 10^{-4} Торр.

Уменьшение линейных размеров микросхем и повышение уровня их интеграции заставляют проектировщиков искать средства борьбы с потребляемой W_n и рассеиваемой W_p мощностью. При сокращении линейных размеров микросхем в два раза их объемы изменяются в восемь раз. Пропорционально этим цифрам должны меняться и значения W_n и W_p , в противном случае схемы будут перегреваться и выходить из строя [12].

Известно, что $W = UI$. Напряжение питания современных микросхем составляет 5 – 3 В. Появились схемы с напряжением питания 2,8 В, что выходит за рамки принятых стандартов. Дальнейшее понижение напряжения нежелательно, так как всегда в электронных схемах должно быть обеспечено необходимое соотношение сигнал-шум, гарантирующее устойчивую работу ЭВМ.

Протекание тока по микроскопическим проводникам сопряжено с выделением большого количества тепла. Поэтому, создавая сверхбольшие интегральные схемы, проектировщики вынуждены снижать тактовую частоту работы микросхем. Использование максимальных частот работы возможно только в микросхемах малой и средней интеграции. Максимальная частота $f_{\max} = 10^{11} - 10^{12}$ Гц доступна очень немногим материалам: кремнию Si, арсениду галлия GaAs и некоторым другим. Поэтому они чаще всего и используются в качестве подложек в микросхемах. Неизбежно и применение новых материалов для полупроводниковых транзисторов будущего. Прежде всего, речь идет о новом диэлектрике затвора транзистора с высокой диэлектрической проницаемостью (так называемый «*high-k*»-материал), который уже применяется взамен используемого в настоящее время диоксида кремния (SiO_2), а также о новых металлических сплавах, совместимых с новым диэлектриком затвора.

Вместо алюминия для проводников все чаще применяется медь – материал с меньшим удельным сопротивлением. Современный уровень развития нанотехнологий в микроэлектронике позволяет создавать пластины и даже трубки толщиной в атомный слой, так что возможности этого направления почти безграничны. Они и служат гарантией того, что закон Мура будет соблюдаться еще многие годы.

Массовое распространение персональных ЭВМ изменило требования к программам. Главными из этих требований стали: простота правил работы, эстетичность, надежность программ, универсальность их функций, простота обучения работе на компьютерах.

Переход к конструированию ЭВМ на ультраСБИС должен сопровождаться снижением тактовой частоты работы схемы. Это может быть обеспечено либо за счет архитектурных решений, либо за счет новых принципов построения и работы микросхем, например параллельной работы. Решаемые задачи должны быть существенно распараллелены – в частности, разделены на подзадачи, которые могут выполняться одновременно на множестве ядер [18]. Сегодняшние одноядерные и многоядерные процессоры способны одновременно обслуживать всего несколько потоков. Будущие процессоры смогут обрабатывать множество потоков – сотни, а в некоторых случаях даже тысячи.

Дальнейшие достижения в микроэлектронике связываются с электронной (лазерной), молекулярной, ионной и рентгеновской литографией.

Большие исследования проводятся также в области использования явления сверхпроводимости и туннельного эффекта – эффекта Джозефсона. Работа микросхем при температурах, близких к абсолютному нулю (~ 273 °С), позволяет достигнуть f_{\max} , при этом $W_p = W_n = 0$. Очень интересны результаты по использованию "теплой сверхпроводимости". Оказывается, что для некоторых материалов, в частности для солей бария, явление сверхпроводимости наступает уже при температурах около -150 °С. Высказывались соображения, что могут быть получены материалы, имеющие сверхпроводимость при температурах, близких к комнатной.

В качестве еще одного из альтернативных путей развития элементной базы ЭВМ будущих поколений следует рассматривать и биомолекулярную технологию. В настоящее время имеются опыты по синтезу молекул на основе их стереохимического генетического кода, способных менять ориентацию и реагировать на ток, свет и т.п.

Внедрение новых технологий производства микропроцессоров испытывает и экономические проблемы. Например, строительство нового завода по производству микросхем с 0,13-микронной технологией обходится от 2 до 4 млрд дол. Это заставляет искать новые альтернативные пути построения будущих ЭВМ. Интенсивные поиски идут по многим направлениям. Наиболее перспективными из них следует считать:

- создание молекулярных и биокомпьютеров (нейрокомпьютеров);
- разработку квантовых компьютеров;
- разработку оптических компьютеров.

Укажем основные принципы их построения:

Молекулярные компьютеры. Во многих странах проводятся опыты по синтезу молекул на основе их стереохимического генетического кода, способных менять ориентацию и реагировать на воздействия ток, светом и т.п. Например, ученые фирмы *Hewlett-Packard* и Калифорнийского университета (*UCLA*) доказали принципиальную возможность создания молекулярной памяти ЭВМ на основе молекул роксана. Продолжаются работы по созданию логических схем, узлов и блоков. По оценкам ученых, подобный компьютер в 100 млрд раз будет экономичнее современных микропроцессоров.

Уже сейчас на основе углеродных нанотрубок создаются электронные устройства нанометрового (молекулярного) размера. Ожидается, что в обозримом будущем они заменят элементы аналогичного назначения в компьютерах. В результате будет достигнут теоретический предел плотности записи информации (порядка одного бита на молекулу), и вычислительные машины обретут практически неограниченную память и быстроедействие, лимитируемое только временем прохождения сигнала через прибор.

Биокомпьютеры или нейрокомпьютеры. Идея создания подобных компьютеров базируется на основе теории перцептрона – искус-

ственной нейронной сети, способной обучаться. Автором этих идей был Ф. Розенблат. Он указал, что структуры, обладающие свойствами мозга и нервной системы, позволяют получить целый ряд преимуществ:

- параллельность обработки информационных потоков;
- способность к обучению и настройке;
- способность к автоматической классификации;
- более высокую надежность;
- ассоциативность.

Компьютеры, состоящие из нейроподобных элементов, могут искать нужные решения посредством самопрограммирования на основе соответствия множеств входных и выходных данных. В настоящее время уже созданы и используются программные нейропакеты, которые доказывают возможность построения подобных машин на СБИС [10].

Квантовые компьютеры. Принцип работы элементов квантового компьютера основан на способности электрона в атоме иметь различные уровни энергии: E_0, E_1, \dots, E_n . Переход электрона с нижнего энергетического уровня на более высокий связан с поглощением кванта электромагнитной энергии – фотона. При излучении фотона осуществляется обратный переход. Всеми подобными переходами можно управлять, используя действие электромагнитного поля от атомного или молекулярного генератора. Этим исключаются спонтанные переходы с одного уровня на другой. Основным же строительным блоком квантового компьютера служит *qubit* – *Quantum Bit*, который может иметь большое число состояний. Для таких блоков определен логически полный набор элементарных функций.

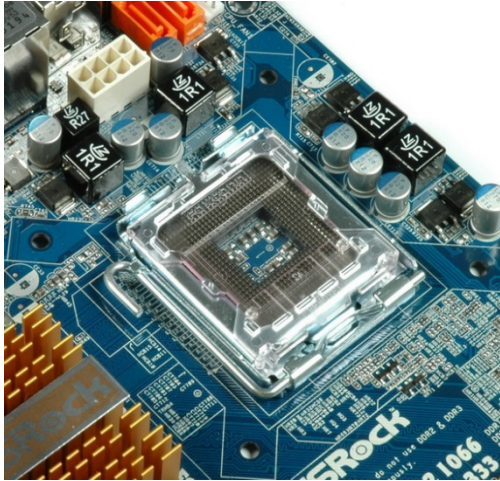
Оптические компьютеры. Идея построения оптического компьютера давно волнует исследователей. Многие устройства ЭВМ используют оптику в своем составе: сканеры, дисплеи, лазерные принтеры, оптические диски *CD-ROM* и *DVD-ROM*. Появились и успешно работают оптоволоконные линии связи. Остается создать устройство обработки информации с использованием световых потоков. Способность света параллельно распространяться в пространстве дает возможность создавать параллельные устройства обработки. Это позволило бы на много порядков ускорить быстроедействие ЭВМ.

Таким образом, можно сделать вывод, что в настоящее время возможности микроэлектроники еще не исчерпаны, но давление пределов уже ощутимо. Основой для ЭВМ будущих поколений будут СБИС совместно с ультраСБИС различной физической природы [16]. При этом структуры ЭВМ и ВС будут широко использовать параллельную работу микропроцессоров.

Вопросы к компьютерному тестированию

1. Какая из структурных единиц ЭВМ предназначена для обработки единичных электрических сигналов?
2. Какая из структурных единиц ЭВМ обеспечивает одновременную обработку группы сигналов?
3. Какая из структурных единиц ЭВМ реализует функционально обособленную часть машинных операций?
4. Как называют схему, у которой выходные сигналы определяются совокупностью входных сигналов?
5. Как называют схемы, у которых единичный сигнал, формирующийся на одном из m выходов, однозначно соответствует комбинации входных сигналов?
6. Как называют схемы, у которых по номеру входного сигнала формируется однозначная комбинация выходных сигналов?
7. Как называют комбинационные схемы, у которых осуществляется коммутация $2m$ входных сигналов на один выход?
8. Как называют комбинационные схемы, у которых один вход подключен к нескольким выходам?
9. Какую логическую зависимость реализует схема сравнения (компаратор)?
10. Какую логическую зависимость реализует комбинационный полусумматор?
11. Какую логическую зависимость реализует многоразрядный комбинационный сумматор?
12. Приведите числовые значения для S и P в строках таблицы истинности полусумматора.

13. Приведите числовые значения для S и P в строках таблицы истинности полного сумматора.
14. Как называют схему, у которой выходные сигналы определяются совокупностью входных сигналов и состояний схем памяти?
15. К устройствам какого типа относят сумматоры?
16. К устройствам какого типа относят регистры сдвига?
17. Что означает прописная буква R (S , T , D) в названии триггера?
18. Какое число сигналов позволяет подсчитывать счетчик, состоящий из пяти триггеров?
19. Какое число сигналов позволяет подсчитывать счетчик, состоящий из одного триггера?
20. Какое число триггеров потребуется при реализации делителя на 5?
21. Какое число триггеров потребуется при реализации делителя на 32?
22. Какой тип литографии в технологии изготовления ИС в настоящее время доминирующий?
23. Во сколько раз, по закону Мура, каждые полтора-два года будет увеличиваться число интегральных транзисторов в кристаллах микропроцессоров?
24. Что является причиной невозможности повышения тактовой частоты работы при создании сверхбольших интегральных схем?
25. Какие пути используют для снижения тактовой частоты работы схем при переходе к конструированию ЭВМ на ультраСБИС?
26. С каким физическим явлением связан эффект Джозевсона, используемый при создании новых технологий производства СБИС?
27. Какой из перечисленных материалов обладает свойством «теплой сверхпроводимости»?
28. Какое из направлений создания будущих ЭВМ связано с использованием углеродных нанотрубок?
29. Какое из направлений создания будущих ЭВМ базируется на основе теории перцептрона?
30. Какое из направлений создания будущих ЭВМ основано на способности электрона в атоме иметь различные уровни энергии?



Глава 5

ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИИ ЭВМ

Рассматриваемые вопросы:

- 5.1. Общие принципы функциональной и структурной организации ЭВМ.
- 5.2. Совместимость ЭВМ.
- 5.3. Функциональные компоненты ЭВМ и их взаимосвязь.
- 5.4. Центральные устройства ЭВМ.
 - 5.4.1. Системная (материнская) плата.
 - 5.4.2. Микропроцессорный набор.
 - 5.4.3. Узлы, выполняющие служебные функции.

5.1. Общие принципы функциональной и структурной организаций ЭВМ

Согласно классической фон-неймановской структуре, преобладавшей в первом и втором поколениях, центром архитектуры ЭВМ являлся процессор (рис. 5.1).

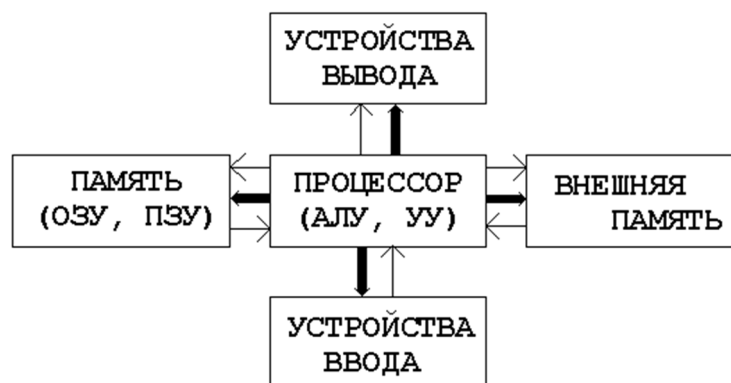


Рис. 5.1. Функциональное устройство ЭВМ 1-го и 2-го поколений (жирными стрелками показаны управляющие воздействия, а тонкими – направления информационных потоков)

Из приведенной схемы отчетливо видно, что центр такой конструкции – процессор. Во-первых, он управляет всеми устройствами, а во-вторых, через него проходят все информационные потоки. Описанной системе по определению присущ принципиальный недостаток – процессор оказывается чрезмерно перегруженным. Полностью регулируя обмен между всеми устройствами, он часто вынужден пассивно ожидать окончания ввода с медленных (как правило, содержащих механические части) устройств, что существенно снижает эффективность работы всей системы в целом.

Возникшее противоречие между постоянно растущей производительностью процессора и относительно низкой скоростью обмена с внешними устройствами стало отчетливо заметно уже во время расцвета вычислительной техники второго поколения. Поэтому при проектировании следующих поколений инженеры начали принимать специальные меры для "разгрузки" процессора и его освобождения от детального руководства вводом/выводом [15].

Переход к третьему и далее к четвертому поколению ЭВМ не только сопровождался многократным повышением плотности монтажа в микросхемах, но и изменением общей стратегии применения вычислительной техники. На смену громоздким ЭВМ коллективного пользования пришли персональные компьютеры, предназначенные прежде всего для индивидуальной работы отдельных пользователей. Архитектура при этом продолжила свое развитие и совершенствование в направлении освобождения процессора от руководства процессами ввода/вывода. В результате современный ПК приобрел структуру, приведенную на рис. 5.2.

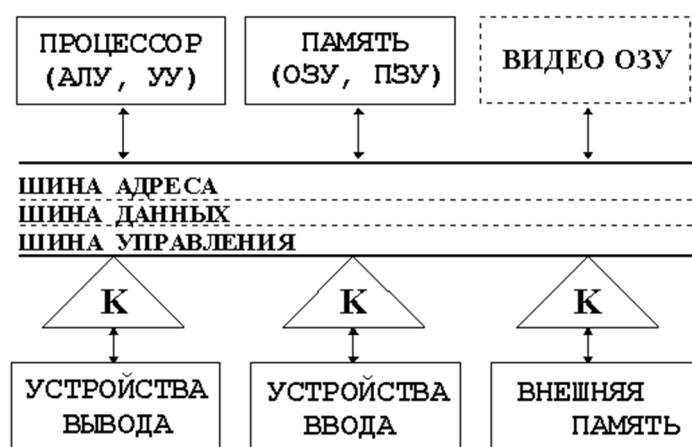


Рис. 5.2. Функциональное устройство ПК 4-го поколения

Главная особенность такой схемы – наличие выделенной *шины (магистралей)* для передачи информации между функциональными узлами компьютера. Она состоит из трех частей:

- *шина адреса*, определяющая, куда именно направляется информация по шине;
- *шина данных*, по которой передается информация;
- *шина управления*, определяющая особенности обмена и синхронизирующая его.

К шине подсоединяются все устройства компьютера, начиная от процессора и кончая устройствами ввода и вывода. Существенная особенность архитектуры ПК – наличие специализированных процессоров ввода/вывода, которые называются *контроллерами* (на схеме обозначены К). Их роль заключается в поддержке процессов обмена информацией для данного устройства, а также в согласовании со стандартной шиной всевозможных внешних устройств различных производителей.

При компоновке ЭВМ выделяют следующие структурные единицы: устройства, узлы, блоки и элементы. Нижний уровень обработки информации реализуют элементы. Каждый элемент предназначен для обработки единичных электрических сигналов, соответствующих битам информации.

Как уже было отмечено в предыдущей главе, узлы обеспечивают одновременную обработку группы сигналов, блоки реализуют функционально обособленную часть машинных операций (блок выборки команд, блок записи-чтения и др.), устройства предназначаются для выполнения отдельных машинных операций и их последовательностей.

Элементы ЭВМ можно классифицировать по различным признакам, наиболее часто такими являются: тип сигнала, назначение элементов, технология их изготовления и т.д.

По своему назначению элементы делятся на формирующие, логические и запоминающие.

К формирующим элементам относятся различные формирователи, усилители, усилители-формирователи и т. д. Данные элементы служат для выработки определенных электрических сигналов, восстановления их параметров (амплитуды, полярности, мощности, длительности).

Логические элементы преобразуют входные сигналы в соответствии с элементарными логическими функциями на основе алгебры логики.

Запоминающим элементом называется элемент, который способен принимать и хранить код двоичной цифры (единицы или нуля).

Вспомним элементную базу ЭВМ, подробно рассмотренную в предыдущей лекции.

Дешифраторы – это комбинационные схемы с n входами и $m = 2^n$ выходами. Дешифраторы широко используются в ЭВМ для выбора информации по определенному адресу, расшифровки кодов операции и др.

Шифраторы решают задачу, обратную схемам ДПТ, то есть по номеру входного сигнала формируют однозначную комбинацию входных сигналов.

Схемы сравнения, или компаратор, обычно строятся как поразрядные для сравнения данных.

Комбинационный сумматор, принципы работы и построения которого вытекают из правил сложения двоичных цифр.

Более сложным преобразователем информации являются схемы с памятью. Наличие памяти в схеме позволяет запоминать промежуточные состояния обработки и учитывать их значения в дальнейших преобразованиях. В качестве простейшего запоминающего элемента в современных ЭВМ используют триггеры.

Регистром называется узел, предназначенный для машинного приема, временного хранения и выдачи машинного слова (группа бит). Они представляют собой совокупность триггеров, число которых соответствует числу рядов в слове.

Счетчик – это узел ЭВМ, позволяющий подсчитывать поступающие на его вход сигналы и фиксировать результаты в виде многозначного двоичного числа. Они используются для подсчета импульсов, сдвигов, формирования адресов и т.д.

Сумматор – это узел ЭВМ, в котором суммируются коды чисел. Он представляет собой комбинацию одноразрядных сумматоров, состоящих из сумматоров комбинационного типа.

ЭВМ включают, кроме аппаратной части и программного обеспечения, большое количество функциональных средств.

Считается, что коды, система команд, алгоритмы выполнения различных процедур и взаимодействия аппаратной части и программного

обеспечения, способы использования устройств при организации их совместной работы, составляющие принципы функционирования ЭВМ, образуют *функциональную организацию ЭВМ*.

Способы реализации функций ЭВМ определяют структурную организацию ЭВМ. Реализованы принципы функционирования ЭВМ могут быть по-разному: аппаратными, программно-аппаратными или программными средствами.

Элементная база, функциональные узлы и устройства ЭВМ, программные модули различных видов (обработчики прерываний, драйверы, программы, файлы и др.) являются структурными компонентами. Они определяют способы реализации функций ЭВМ и составляют *структурную организацию ЭВМ*.

5.2. Совместимость ЭВМ

Серьезным является вопрос совместимости ЭВМ. Совместимость – это приспособленность к работе с одними и теми же программами (*программная совместимость*, или *TWAIN-совместимость*) и получению одних и тех же результатов при обработке одной и той же однотипно представленной информации (*информационная совместимость*). Если аппаратная часть электронных вычислительных машин допускает их электрическое соединение для совместной работы и предусматривает обмен одинаковыми последовательностями сигналов, то имеет место и *техническая совместимость ЭВМ*.

Совместимым ЭВМ необходимо иметь одинаковую функциональную организацию: информационные элементы (символы) должны одинаково представляться при вводе и выводе из ЭВМ, система команд должна обеспечивать в этих ЭВМ получение одинаковых результатов при одинаковых преобразованиях информации. К работе таких машин предъявляют требования функционально совместимых операционных систем (а для этого должны быть совместимы методы и алгоритмы планирования и управления работой аппаратурно-программного вычислительного комплекса). Аппаратурные средства имеют согласованные питающие напряжения, частотные параметры сигналов, а главное – состав, структуру и последовательность выработки управляющих сигналов.

При неполной совместимости ЭВМ (при наличии различий в их функциональной организации) применяют эмуляторы, т.е. программные преобразователи функциональных элементов.

5.3. Функциональные компоненты ЭВМ и их взаимосвязь

Структура ЭВМ – это некоторая модель, устанавливающая состав, порядок и принципы взаимодействия входящих в нее компонентов. Функциональная схема современной ЭВМ представлена на рис. 5.3.



Рис. 5.3. Функциональная схема современной ЭВМ

Рассмотрим состав и принцип взаимодействия основных компонентов ЭВМ.

Системная (материнская) плата – важнейший элемент ПК. На ней размещаются устройства, непосредственно осуществляющие процесс обработки информации (вычислений). Как правило, это микропроцессор, внутренняя память, системная шина, контроллер клавиатуры и прерываний, генератор тактовой частоты (ГТИ), таймер и др. Схемы, управляющие другими внешними устройствами компьютера, как правило, находятся на отдельных платах, вставляемых в унифицированные разъемы (слоты) на материнской плате. Через эти разъемы контроллеры устройств подключаются непосредственно к системной магистрали передачи данных в компьютере – шине. Иногда эти контроллеры могут располагаться на системной плате. Наборы микросхем, на основе которых исполняются системные платы, называют чипсетами.

Все дополнительные устройства взаимодействуют с процессором и оперативной памятью через системную магистраль передачи данных –

шину. Данные могут передаваться между внешними устройствами и процессором, оперативной памятью и процессором, внешними устройствами и оперативной памятью или между устройствами ввода-вывода. Шина характеризуется типом, разрядностью, частотой и количеством подключаемых внешних устройств. При работе с оперативной памятью шина проводит поиск нужного участка памяти и обменивается информацией с найденным участком. Эти задачи выполняют две части системной шины: адресная шина и шина данных. Аппаратно-логические устройства, отвечающие за совместное функционирование различных компонентов, называют интерфейсами. Современный компьютер заполнен разными интерфейсами, обеспечивающими всеобщее взаимодействие. На интерфейсы существуют стандарты.

Совокупность интерфейсов, реализованных в ЭВМ, образует то, что называют архитектурой ЭВМ [2]. Для добавления нового дополнительного устройства необходим контроллер – устройство, аппаратно согласовывающее работу системы и дополнительного устройства. Кроме того, необходим драйвер этого устройства – программа, позволяющая программно связать это устройство с системой в целом. Контроллер должен учитывать аппаратные особенности подключаемого устройства, а драйвер должен позволить операционной системе, используя стандартный набор командных запросов, управлять нестандартным устройством.

Центральной частью ЭВМ является системный блок, с присоединенными к нему клавиатурой, монитором и мышью. В системном блоке располагаются все основные устройства: микропроцессор – мозг ЭВМ, который выполняет поступающие на его вход команды: проводит вычисления и управляет работой остальных устройств; оперативная память, предназначенная для временного хранения программ и данных; контроллеры, предназначенные для независимого от процессора управления отдельными процессами в работе ЭВМ; накопители на гибких магнитных дисках, используемые для чтения и записи на дискеты; накопитель на жестком магнитном диске, предназначенный для чтения и записи на жесткий магнитный диск (винчестер); дисководы для компакт-дисков, обеспечивающие возможность чтения и записи информации; блок питания, преобразующий электропитание сети в постоянный ток, подаваемый на электронные схемы ЭВМ; счетчик времени и другие устройства.

Все компоненты ЭВМ по их функциональному отношению к работе с информацией можно условно разделить: на устройства обработки информации (центральный процессор, специализированные процессоры); устройства хранения информации (жесткий диск, CD-ROM, оперативная память, др.); устройства ввода информации (клавиатура, мышь, микрофон, сканер и т.д.); устройства вывода информации (монитор, принтер, акустическая система и т.д.).

Микропроцессор (МП), или центральный процессор (*CPU*), – основной рабочий компонент компьютера, который выполняет арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера.

Генератор тактовых импульсов генерирует последовательность электрических импульсов, частота которых определяет тактовую частоту машины, т.е. 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.

Память (внутренняя – системная, включающая оперативное запоминающее устройство (ОЗУ) и постоянное запоминающее устройство (ПЗУ), и внешняя дисковая). ПЗУ служит для хранения неизменяемой (постоянной) программной и справочной информации. ОЗУ предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе. Дисковая память относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач.

Таймер. Это внутримашинные электронные часы, обеспечивающие при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд).

Внешние устройства (ВУ). От их состава и характеристик во многом зависят возможность и эффективность применения ЭВМ. ВУ

обеспечивают взаимодействие машины с окружающей средой: пользователями, объектами управления и другими ЭВМ. К ним относят: внешние запоминающие устройства (ВЗУ), или внешнюю память ПК; устройства ввода информации; устройства вывода информации; средства связи и телекоммуникации.

Монитор – 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.

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); сканеры (читающие автоматы) – для автоматического считывания с бумажных носителей и ввода машинописных текстов, графиков, рисунков, чертежей; манипуляторы (устройства указания); сенсорные экраны – для ввода отдельных элементов изображения, программ или команд с полиэкрана дисплея в ЭВМ.

К устройствам вывода информации относятся: принтеры – печатающие устройства для регистрации информации на бумажный носитель; графопостроители (плоттеры) – для вывода графической информации (графиков, чертежей, рисунков) из ПК на бумажный носитель.

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

Дополнительные схемы: математический сопроцессор, сопроцессор ввода-вывода, контроллер прерываний и др. [20].

5.4. Центральные устройства ЭВМ

Как уже было сказано выше, в *состав центральных устройств ЭВМ* входят: центральный процессор, основная память, универсальные интерфейсы и ряд дополнительных узлов, выполняющих служебные функции. Все это, как правило, располагается на материнской плате.

5.4.1. Системная (материнская) плата

Важнейший узел компьютера – системная плата (*system board*), иногда называемая материнской (*motherboard*), основной, или главной платой (*main board*); все эти термины взаимозаменяемы.

Практически все внутренние компоненты персонального компьютера вставляются в материнскую плату, и именно ее характеристики определяют возможности компьютера, не говоря уже о его общей производительности.

Материнская плата – вершина современной технологии изготовления печатных плат. Количество слоев диэлектрического материала и соответственно уровней, на которых располагается сетка различных проводников и шин, более восьми. В обозначении материнской платы всегда указывают формат – форм-фактор (*form factor*), которым обозначается физическая конструкция и типоразмер системных плат и компьютерных корпусов.

Основные стандартные форм-факторы (размеры) системных плат: *ATX* – самый распространенный полноразмерный формат материнской платы (30,5 × 24,4 см), является оптимальным для стационарного компьютера, так как размеры платы позволяют не особо экономить и удобно развести и разместить на ней все элементы, устанавливается в большинство корпусов для стационарных ПК.

Mini-ATX – очень распространенный урезанный формат материнской платы (28,4 × 20,8 см), типичен для бюджетных материнских плат, обычно имеет меньшее количество разъемов (например, всего 2 слота для памяти), устанавливается в большинство корпусов для стационарных ПК и уменьшенных корпусов одноименного формата.

Micro-ATX – чуть менее распространенный уменьшенный формат материнской платы (24,4 × 24,4 см), используется преимущественно для сборки мультимедийных центров и может иметь встроенные беспроводные адаптеры, устанавливается в специальные корпуса для мультимедийных центров.

Flex ATX является расширением к спецификации *Micro ATX*. По площади на 25 – 30 % меньше платы *Micro ATX* и предназначен для установки в нестандартные корпуса вроде *iMAC*.

Существуют и другие менее распространенные форматы:

BTX (Balanced Technology Extended) – стандарт, разработанный с целью эффективного охлаждения внутренних компонентов системного блока миниатюрных компьютеров. Платы *BTX* имеют размеры 26,7 × 32,5 см и 7 слотов расширения;

mBTX (micro BTX) – уменьшенный вариант *BTX*, поддерживающий 4 слота расширения. Формат *mBTX* имеет размеры 26,7 × 26,4 см;

mini-ITX – стандарт электрически и механически совместимый с форм-фактором *ATX*. Форм-фактор *mini-ITX* разработан компанией *VIA Technologies* и имеет небольшие размеры (17 × 17 см).

Серверные форматы:

EATX (Extended ATX) – основное отличие от *ATX* – это размеры (30,5 × 33,0 см);

SSI EEB (Server Standards Infrastructure Entry Electronics Bay) – размеры 30,5 × 33,0 см. Основной разъем для подключения блока питания имеет 24 + 8 контактов.

SSI CEB (SSI Compact Electronics Bay) – данный форм-фактор имеет 24 + 8 контактов в основном разьеме. Габаритные размеры таких плат – 30,5 × 25,9 см.

Фактически материнская плата – это большая коллекция внешних и внутренних разъемов, предназначенных для установки тех или иных комплектующих, связанных между собой с помощью шины. На рис. 5.4 и 5.5 изображены внешние и внутренние разъемы и элементы современной материнской платы.

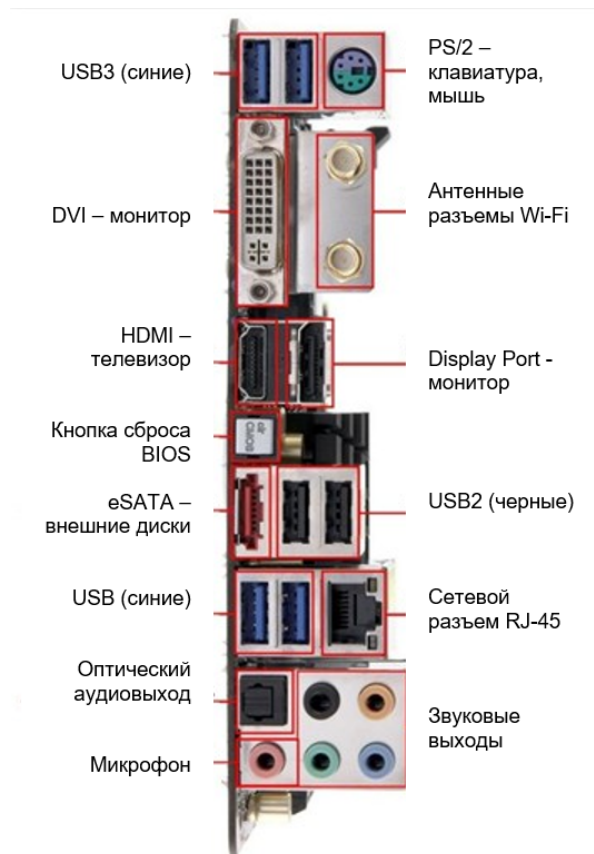


Рис. 5.4. Внешние разъемы современной материнской платы

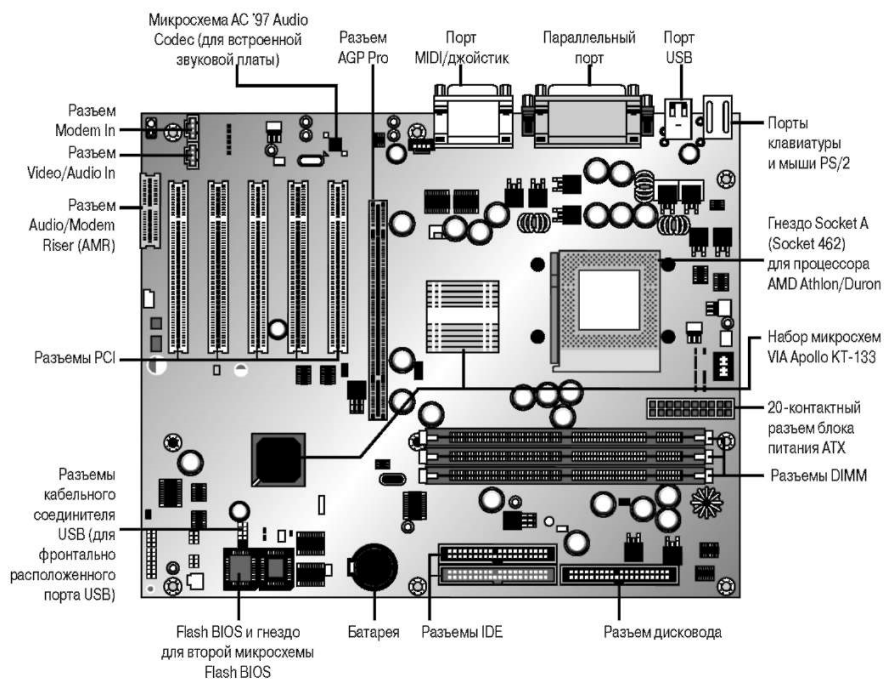


Рис. 5.5. Внутренние разъемы и элементы современной материнской платы

Процессорный разъем (слот)

Процессорный разъем, или, как его еще называют, *Socket* (слот), является местом соединения процессора и материнской платы (см. рис. 5.5). Процессорные разъемы маркируются либо по количеству ножек в разьеме, либо по маркировке линейки процессоров.

Технологический процесс в настоящее время идет очень быстро, меняются процессоры, меняются процессорные разъемы.

Устаревающие слоты процессоров *Intel* имеют маркировку: *LGA 775*.

Современные слоты процессоров *Intel* имеют маркировку: *LGA 1150*, *LGA 1155*, *LGA 2011*.

Устаревающие слоты процессоров *AMD* имеют маркировку: *AM3*, *AM3+*, *FM1*.

Современные слоты процессоров *AMD* имеют маркировку: *FM2*, *FM2+*, *FX*.

Главное то, что на процессоре и материнской плате разъемы должны быть одинаковыми, иначе процессор просто не установится.

Разъемы (слоты) памяти

На недорогих материнских платах часто выполняют только два слота для установки модулей памяти. Материнские платы среднего и выше ценового диапазона обычно имеют четыре и более слотов памяти (см. рис. 5.5). Это позволяет не только установить больший ее объем, но и без проблем добавить память в будущем без потери в производительности за счет использования двухканального режима.

Старые материнские платы использовали память *DDR*, *DDR2* и более старого типа. Ходовыми в настоящее время являются память типа *DDR3* и *DDR4*.

У каждой материнской платы есть ограничение на максимальный объем модуля памяти, который можно установить в слот, а также ограничение на общий поддерживаемый объем памяти.

Разъемы (слоты) для установки видеокарт

Старые материнские платы имели разъем *AGP* для установки видеокарт, все современные материнские платы имеют разъем *PCI Express* (*PCI-E*) (см. рис. 5.5). Этот разъем имеет три ревизии (*PCI-E v.1*, *v.2* и *v.3*), которые отличаются пропускной способностью шины (скоростью

передачи данных между материнской платой и видеокартой). Эти ревизии совместимы, кроме того в зависимости от пропускной способности разъемы также маркируются как $x4$, $x8$, $x16$.

Материнские платы среднего и высшего класса могут иметь от 1 до 4 разъемов *PCI-E*, и соответственно, на них можно устанавливать несколько видеокарт в режиме *SLI* (для видеокарт *NVidia*) или *CrossFire* (для видеокарт *AMD*). В этом случае более дешевые материнские платы поддерживают работу нескольких видеокарт только на пониженной скорости шины (например: $x8 + x8$ или $x4 + x4 + x4$). Дорогие игровые видеокарты обеспечивают спаренный режим работы $x16 + x16$ или $x8 + x8 + x8 + x8$, что повышает быстродействие видеосистемы при установке нескольких мощных видеокарт.

Основная системная шина

Пропускная способность основной системной шины *HT* (*Hyper Transport*), а ранее *FSB* (*Front System Bus*) характеризуется в основном скоростью обмена данными между процессором и оперативной памятью. Современные материнские платы имеют пропускную способность шины 5200 Мб/с (5,2 Гб/с) и выше, уже ушедшее их поколение имело шину 1000 – 2000 Мб/с (1 – 2 Гб/с).

Встроенное (интегрированное) видео

В старых материнских платах с интегрированной видеокартой видеочип распаивался на самой плате. В современных материнских платах этого чипа нет, так как уже многие современные процессоры имеют видеоядро. В любом случае встроенное видео не имеет своей отдельной памяти и использует для работы оперативную память компьютера. На материнских платах с интегрированной видеокартой есть разъемы для подключения монитора и/или телевизора. На старых материнских платах это разъем *VGA*, на новых – *DVI* и *HDMI* (см. рис. 5.4).

Встроенный (интегрированный) звук

Все современные материнские платы имеют аудиокодек класса *HDA* (*High Definition Audio*), который находится уже в чипсете. Такой аудиокодек выдает качество звука, сопоставимое с бюджетными дискретными звуковыми картами, и его звучание больше зависит от качества подключенной акустической системы (колонок).

Материнские платы обычно имеют три или шесть гнезд диаметром 3,5 мм для подключения аудиоустройств, одно из которых всегда задействовано для подключения микрофона (см. рис. 5.4).

Также материнская плата может иметь цифровые выходы аудиосигнала – коаксиальный (в старых платах) и оптический (в новых) (см. рис. 5.4). При этом искажения звука при передаче сигнала от звуковой карты к колонкам практически отсутствуют.

Встроенная (интегрированная) сетевая карта

Сетевая карта входит в состав чипсета и пользуются ресурсами компьютера. Почти все современные материнские платы имеют сетевую карту со скоростью передачи данных 1000 Мб/с (1 Гб/с).

Материнская плата со встроенной сетевой картой имеет разъем *RJ-45* (см. рис. 5.4). Бывают материнские платы с двумя сетевыми картами. В таком случае они имеют два таких разъема.

Поддержка различных носителей

Современные материнские платы имеют полностью совместимые *разъемы SATA2* (3 Гб/с) и *SATA3* (6 Гб/с) для подключения различных носителей данных: жестких дисков (*HDD*), оптических приводов (*ODD*), быстрых твердотельных накопителей (*SSD*) (см. рис. 5.5).

Разъем IDE для подключения дисков старого типа морально устарел и есть далеко не на всех современных материнских платах.

Современные HDD и ODD имеют интерфейсные разъемы *SATA2* или *SATA3*, и это не принципиально, так как скорость у них ограничивается механикой (до 150 Мб/с), тогда как интерфейс позволяет передавать данные со скоростью на порядок выше.

Современные SSD диски имеют интерфейс *SATA3* и могут подключаться к разъемам *SATA2*, но иногда бывают проблемы с совместимостью и падения скорости. В таком случае все-таки желательно иметь на материнской плате разъемы *SATA3*.

Современные HDD объемом 3 Гб и более не поддерживаются материнскими платами с классическим *BIOS*. В этом случае система просто не видит весь объем диска. Если планируется использовать жесткие диски такого объема, то материнская плата должна поддерживать интерфейс *UEFI*.

Разъемы для плат расширения

Старые материнские платы использовали разъемы *PCI* для установки плат расширения. Такой разъем может понадобиться, если имеются такие платы, как профессиональная звуковая карта или ТВ-тюнер.

На современных материнских платах для установки плат расширения используются разъемы *PCI Express: PCI-E x1, x2, x4* (см. рис. 5.5).

Беспроводные технологии

Некоторые современные материнские платы могут иметь встроенные беспроводные адаптеры *Wi-Fi* и *Bluetooth*. Они могут понадобиться в случае, если эту материнскую плату планируется использовать для сборки мультимедийного центра в компактном корпусе. Для подключения антенны используются специальные гнезда на внешнем разьеме (см. рис. 5.4).

Рассмотрим далее более подробно характеристику компонентов, размещенных на материнской плате.

5.4.2. Микропроцессорный набор

Эффективность работы компьютерной системы и ее возможности во многом определяются установленным на материнской плате микропроцессорным набором, называемым *ChipSet* – **чипсет**. Это своеобразный процессор, который связывает и управляет всеми устройствами платы: центральным процессором, памятью, видеокартой, жесткими дисками, платами расширения, внешними и внутренними разъемами и т.д. От того, насколько мощный и современный чипсет, зависит производительность материнской платы и соответственно всего компьютера.

Внешне микросхемы *Chipset*'а выглядят как самые большие после процессора, с количеством выводов от нескольких десятков до нескольких сотен.

В уже устаревшей схеме построения материнской платы (рис. 5.6) микросхемы чипсета делились на два блока – «северный» и «южный» мосты по их расположению на схеме [14].

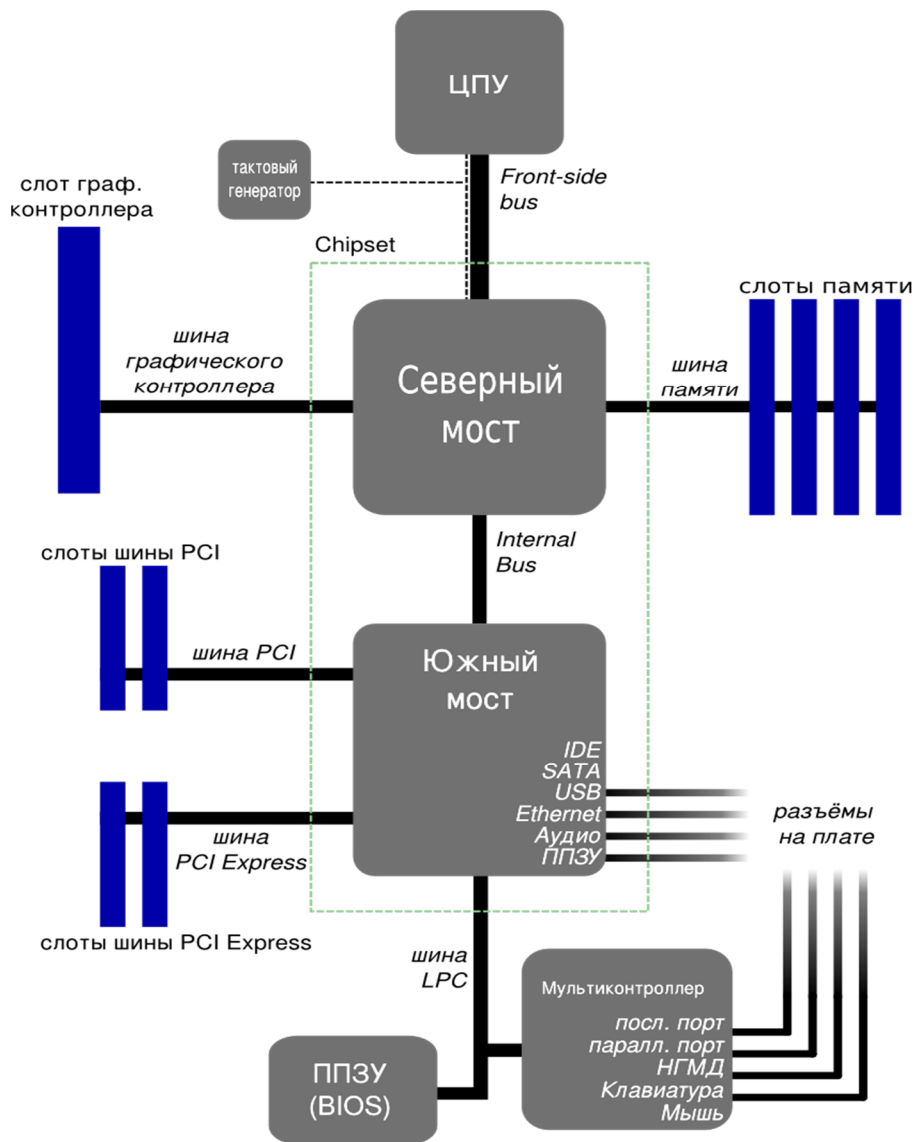


Рис. 5.6. Схема чипсета «северный» и «южный» мосты

Функции «северного» моста – обеспечение работы процессора с оперативной памятью (контроллер *RAM*) и видеокартой (контроллер *PCI-E x16*). «Южный» же отвечает за связь процессора с другими устройствами компьютера – жесткими дисками, оптическими приводами, картами расширения и так далее через контроллеры *SATA*, *IDE*, *PCI-E x1*, *PCI*, *USB*, звук.

Основной характеристикой производительности чипсета в этой архитектуре является шина данных (*System Bus*), предназначенная для обмена информацией между различными составляющими компьютер частями. Все компоненты работают с чипсетом через шины, причем каждый со своей скоростью. Это наглядно видно на схеме чипсетов.

Производительность же всего ПК зависит именно от скорости той шины, которая связывает его с самим чипсетом. В терминологии чипсетов *Intel* эта шина обозначается как *FSB (Front Side Bus)*. В описании материнской платы она именуется «частотой шины», или «пропускной способностью шины». Разберем подробнее эти характеристики шины данных. Она определяется двумя показателями – частотой и шириной.

Частота – это скорость передачи данных, которая измеряется в мегагерцах (МГц, MHz) или гигагерцах (ГГц, GHz). Чем выше этот показатель, тем выше производительность всей системы в целом (например, 3 ГГц).

Ширина – количество байт, которое шина имеет возможность передать за один раз в байтах (например, 2 Бт). Чем больше ширина, тем большее количество информации шина сможет передать за определенный промежуток времени.

При умножении двух этих величин мы получаем третью, которая как раз и указывается на схемах – пропускная способность, которая измеряется в гигабайтах в секунду (Гб/с, Gb/s). Из нашего примера – производим умножение 3 ГГц на 2 байта и получаем 6 Гб/с.

Связь же «северного» моста с оперативной памятью происходит с помощью встроенного в него двухканального контроллера через шину *RAM Bus*, имеющую 128 контактов ($\times 128$). При работе с памятью в одноканальном режиме задействуются только 64 дорожки, поэтому для максимальной производительности рекомендуется использовать два модуля памяти, подсоединенных на разные каналы.

Архитектура без «северного» моста

В процессорах последнего поколения «северный» мост уже встроен в микросхему самого процессора, что значительно повышает его производительность. Поэтому на новых системных платах он вообще отсутствует – остается только «южный» мост.

На примере ниже (рис. 5.7) в чипсете отсутствует «северный» мост, так как его функцию на себя берет процессор со встроенным видеоядром.

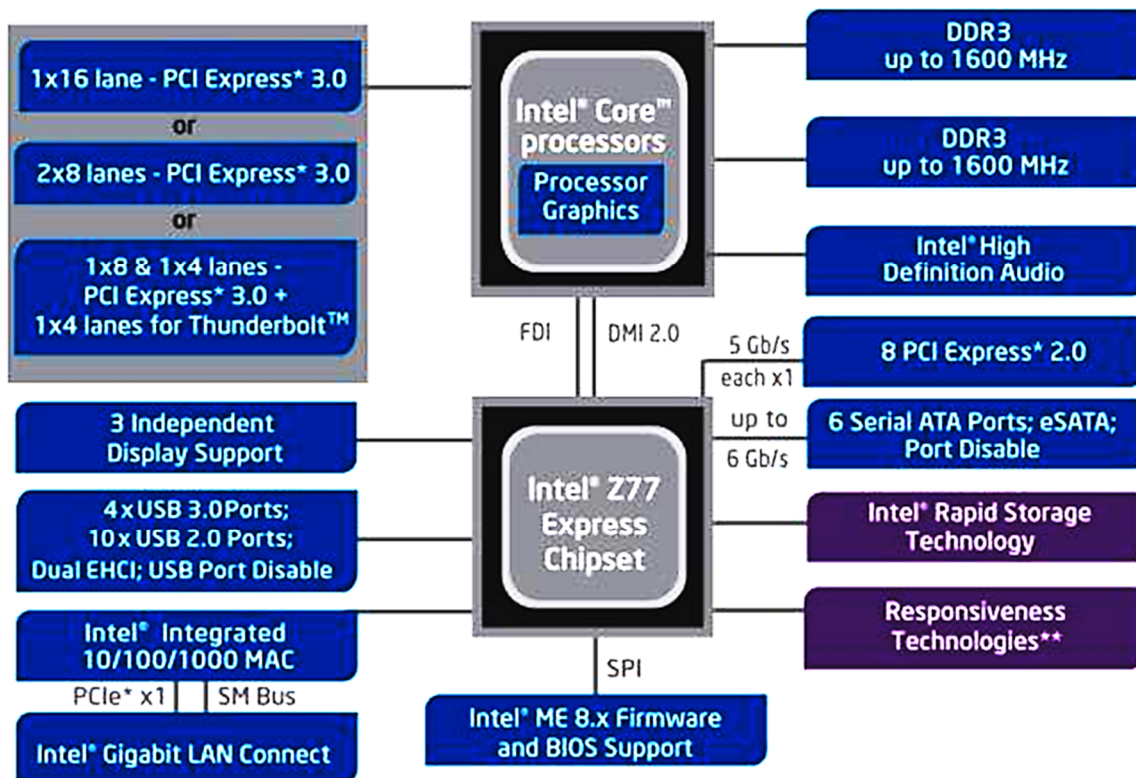


Рис. 5.7. Схема чипсета без «северного» моста

В работе современных процессоров используется шина *QPI* (*QuickPath Interconnect*), а также графический контроллер *PCI-E* x16, который раньше был в «северном» мосту, а теперь встроен в процессор. В результате того, что они стали встроенными, характеристики основной шины данных не так важны, как это было в архитектуре предыдущего поколения с двумя мостами.

В современных чипсетах на новых платах присутствует другой параметр работы шины – трансферы в секунду, который указывает на количество операций по передаче данных в секунду. Например, 3200 МТ/с (мегатрансферы в секунду) или 3,2 ГТ/с (гигатрансферы в секунду).

Эта же характеристика указывается и в описаниях процессоров. Причем, если у чипсета скорость работы шины 3,2 ГТ/с, а у процессора, например 2 ГТ/с, то данная связка будет работать по меньшему значению.

Основными поставщиками на рынке производителей чипсетов являются уже знакомые нам по процессорам фирмы *Intel* и *AMD* (схема чипсета показана на рис. 5.8), а также *Nvidia*, которая больше известна пользователям по своим видеокартам, и *Asus*.

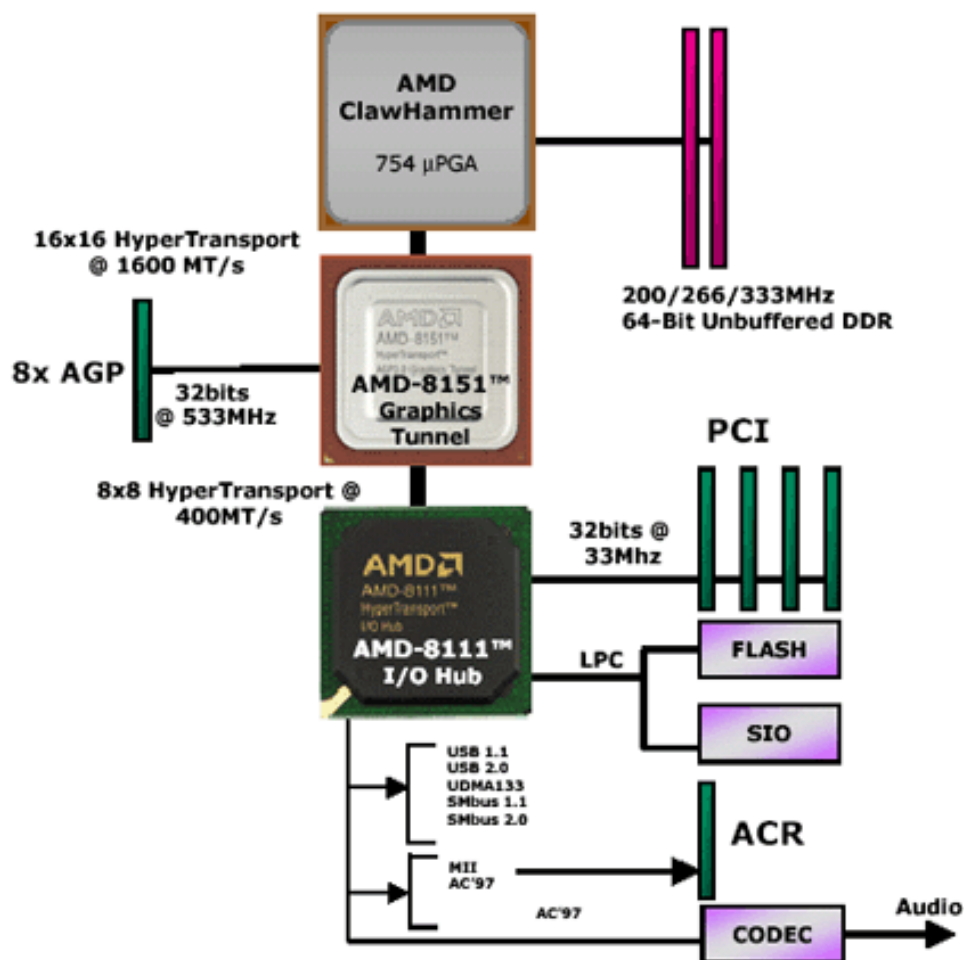


Рис. 5.8. Схема чипсета AMD

Поскольку основными производителями являются на сегодняшний день первые два, рассмотрим современные и уже устаревшие модели.

Чипсеты Intel

Современные с поддержкой встроенной графики – H67, Q67, H77, Z68, Z77, Z87, H87, H81, X79, X99.

Современные без поддержки встроенной графики – P67.

Устаревшие чипсеты – P31, P35, P43, P45, P55, H55, H57, Q57 Q35, G31, G33, G35, G41, G43, G45, X38, X48, X58, X79.

Чипсеты AMD:

- современные чипсеты под *socket FM2*: A85X, A75, A55; под *FM1*: A75, A55;

- устаревшие чипсеты под *socket AM3* и *AM3+*: 990FX, 990X, 970, 890GX, 890FX, 880G, 870, 790GX, 785G, 780V, 780G, 770, 760G;

Чипсеты NVIDIA

Чипсеты *nForce* производства компании *Nvidia* зарекомендовали себя как «прожорливые» в плане энергопотребления, слишком «горячие» в плане тепловыделения, проблемно-ненадёжные в силу вышеперечисленных факторов.

В заключение рассмотрим на примере, как расшифровать спецификацию, указанную в прайс-листах продавца.

Пример*

Материнская плата (мат. плата) *ASUS P8Z77-M Z77, 1155, DDR3 2400(OC)*, 2 × PCIE3.0, 1 × PCIE 2.0 × 16 (4 mode), 1 × PCIe 2.0 × 1, 1 × PCI, 2 × Sata3, 4 × Sata2, RAID 0/1/0+1/5/10, 1×1Gb Lan, SB 8Ch, mATX, BOX.*

"ASUS" – производитель мат. платы *P8Z77-M* – модель (код производителя), причём буква "M" обозначает форм-фактор мат. платы, в данном случае *micro-ATX*;

"1155" – указано, на каком сокете процессора выполнена мат. плата;

"Z77" – указано, на каком чипсете выполнена плата;

"DDR3 2400(OC)*" – какой тип памяти и частоту поддерживает мат. плата, причём (OC) означает *overclock*, т.е. частота 2400 доступна только при разгоне операционной памяти. Частота по умолчанию будет выставлена согласно спецификации процессора;

"2 × PCIE3.0" – указано количество слотов для видеокарт, т.е. 2 шт. *PCI-E ×16* версии *v3.0*;

"1 × PCIE 2.0" – и один слот *PCI-E ×16* версии *v2.0*;

"1 × PCI" – наличие одного слота *PCI* для подключения дополнительных устройств (сетевая, звуковая, *TV*-тюнер);

"2 × Sata3" – указано количество разъёмов *SATA III* с пропускной способностью *6Gb/s*;

"4 × Sata2" – указано количество разъёмов *SATA II* с пропускной способностью *3Gb/s*;

"RAID 0/1/0+1/5/10" – возможность построения *RAID* массива;

"1×1Gb Lan" – наличие встроенной сетевой карты со скоростью 1000 Мбит/с;

"*SB 8Ch*" – наличие встроенной звуковой 8-канальной (7 + 1) карты;

"*mATX*" – форм-фактор мат. платы;

"*BOX*" – полная комплектация мат. платы (коробка, шлейфы, диск с ПО, дополнительные переходники), если "*OEM*", то это только сама мат. плата.

*Источник: <http://expert4help.ru/hard/motherboard/36-vybiraem-materinskiuyu-platu-ili-materinskaya-plata-kak-osnova.html>

5.4.3. Узлы, выполняющие служебные функции

Микросхема **BIOS** (*Basic Input Output System*) – это базовая система ввода/вывода, размещаемая в отдельной микросхеме ПЗУ.

Она представляет собой энергонезависимое постоянное запоминающее устройство, в которое записаны программы, реализующие функции ввода-вывода, а также программа тестирования компьютера в момент включения питания (*POST, Power-On-Self-Test*), и ряд других операционных программ.

В своей работе **BIOS** опирается на сведения об аппаратной конфигурации компьютера, которые хранит еще одна микросхема **CMOS RAM** (*Complementary Metal-Oxide-Semiconductor RAM*). Это энергозависимая память, постоянно подпитываемая от батарейки и располагаемая на системной плате. Кстати, та же батарейка питает и схему кварцевых часов, непрерывно отсчитывающих время и текущую дату. В настоящее время **CMOS RAM** является частью системного чипсета (как правило, «южного» моста).

Во всех современных компьютерах для хранения **BIOS** используются микросхемы на основе *flash*-памяти (*Flash Memory*), которая является одной из разновидностей **EEPROM**. Такая микросхема может быть перезаписана с помощью специальных программ прямо на компьютере. Запись новой версии **BIOS** обычно называется «перепрошивкой». Эта операция может потребоваться, чтобы добавить в код **BIOS** новые функции, исправить ошибки или заменить поврежденный код **BIOS**.

В большинстве случаев *flash*-память устанавливается на панель системной платы, что позволяет при необходимости заменить микросхему, но в некоторых случаях она распаяна прямо на системной плате.

Постепенно все материнские платы переходят с классической прошивки типа *BIOS* на интерфейс *UEFI*, который имеет много дополнительных функций, в частности, поддержку дисков более 2 Тб, установку множества операционных систем на один диск и др.

Основные вопросы, на которые следует обратить внимание при выборе материнской платы ПК.

1. Для какого процессора предназначена материнская плата (*Intel* или *AMD*).
2. Кто производитель материнской платы.
3. Форм-фактор (размер материнской платы) *ATX*, *Mini-ATX*, *Micro-ATX*.
4. Какого класса чипсет установлен на материнской плате.
5. Какой процессорный разъем имеет материнская плата.
6. Какие конкретно модели процессоров поддерживает материнская плата.
7. Тип, объем и частота поддерживаемой памяти.
8. Какие и сколько видеокарт поддерживает материнская плата.
9. Какие интегрированные устройства имеет материнская плата (видео, звук, сетевая карта).
10. Тип прошивки (*BIOS* или *UEFI*).
11. Поддержка типа и объема носителей (*HDD*, *SSD*; *SATA*, *IDE*; меньше 2Гб или больше 2Гб).
12. Количество и расположение разъемов для установки плат расширения.
13. Наличие необходимых внешних разъемов.
14. Размер и расположение радиаторов системы охлаждения.
15. Наличие беспроводных технологий (при необходимости).
16. Срок гарантии на материнскую плату.
17. Дополнительная комплектация (возможность докупить, если чего-то не хватает).

Вопросы к компьютерному тестированию

1. Что является главной отличительной особенностью структуры ПК 4-го поколения и классической фон-неймановской структуры?
2. Из каких частей состоит системная магистраль ПК?
3. Какова основная роль контроллеров ПК?
4. Приведите названия классов элементов при классификации их по назначению.
5. Как называют элемент ЭВМ, предназначенный для ввода, временного хранения и выдачи машинного слова?
6. Как называют элемент ЭВМ, предназначенный для фиксации результатов в виде многоразрядного двоичного числа?
7. Что образует функциональную организацию ЭВМ?
8. Что образует структурную организацию ЭВМ?
9. Что такое *TWAIN*-совместимость ЭВМ?
10. Что такое техническая совместимость ЭВМ?
11. Что такое информационная совместимость ЭВМ?
12. Как называют аппаратно-логическое устройство, отвечающее за совместное функционирование различных компонентов?
13. Как называют программу, позволяющую программно связать устройство с системой в целом?
14. Как называют устройство, аппаратно согласовывающее работу системы и дополнительного устройства?
15. Как называют формат, которым обозначаются физическая конструкция и типоразмер системных плат и компьютерных корпусов?
16. Как называют самый распространенный полноразмерный формат материнской платы, являющийся оптимальным для стационарного компьютера?
17. Какую маркировку имеют современные слоты процессоров *Intel*?
18. Какой тип разъема используется в материнских платах со встроенной сетевой картой?

19. Какой тип разъема используется в материнских платах для установки плат расширения?
20. Как называют процессор, который связывает и управляет всеми устройствами платы?
21. Что относят к функциям «северного» моста схемы чипсета *Intel*?
22. Что относят к функциям «южного» моста схемы чипсета *Intel*?
23. Чему равна пропускная способность шины (Гб/с) при частоте передачи данных 3 ГГц и ширине 2 байта?
24. Почему в современных системных платах отсутствует «северный» мост?
25. Как называют параметр работы шины, который указывает на количество операций по передаче данных в секунду?
26. Как называют показатель, соответствующий количеству байт, которое шина имеет возможность передать за один раз?
27. С какой скоростью (ГТ/с) будет работать связка процессор-чипсет, если у чипсета скорость работы шины 3,2 ГТ/с, а у процессора, например 2 ГТ/с?
28. Как называют энергонезависимое постоянное запоминающее устройство, в которое записаны программы, реализующие функции ввода-вывода, а также программа тестирования компьютера в момент включения питания?
29. В какой микросхеме материнской платы хранятся сведения об аппаратной конфигурации компьютера?
30. Какой тип памяти используют современные компьютеры для хранения *BIOS*?



Глава 6

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР ЭВМ

Рассматриваемые вопросы:

- 6.1. Основные характеристики процессора.
- 6.2. Микропроцессоры типа *CISC*.
- 6.3. Микропроцессоры типа *RISC*.
- 6.4. Микропроцессоры типа *VLIW* и *MISC*.
- 6.5. Физическая и функциональная структура микропроцессора.
- 6.6. Устройство управления.
- 6.7. Арифметико-логическое устройство.
- 6.8. Микропроцессорная память.
- 6.9. Интерфейсная часть ЦП.
- 6.10. Порты ввода-вывода.
- 6.11. Функциональные характеристики ЦП.
- 6.12. Система команд центрального процессора.
- 6.13. Взаимодействие элементов при работе ЦП.
- 6.14. Работа центрального процессора при выполнении задания пользователя.
- 6.15. Работа микропроцессора при выполнении программного прерывания.

6.1. Основные характеристики процессора

На первый взгляд процессор – просто выращенный по специальной технологии кристалл кремния (не зря на профессиональном сленге процессор именуется «камнем»). Однако «камешек» этот содержит в себе множество отдельных элементов – транзисторов (десятки миллионов), которые в совокупности и наделяют компьютер способностью «думать».

Процессор – это центральное устройство компьютера. Он выполняет находящиеся в оперативной памяти команды программы и «общается» с внешними устройствами благодаря шинам адреса, данных и управления, выведенными на специальные контакты корпуса микросхемы.

Выполнение процессором команды предусматривает:

- арифметические действия;
- логические операции;
- передачу управления (условную и безусловную);
- перемещение данных из одного места памяти в другое;
- координацию взаимодействия различных устройств ЭВМ.

Поколение компьютера определяется поколением используемого в нём процессора, поскольку последний является основным его компонентом. Если меняется поколение процессора, значит, меняется и поколение компьютера. Процессор каждого следующего поколения скачкообразно отличался от предыдущего, аккумулируя в себе самые последние на тот период разработки. За всю более чем 35-летнюю историю процессоров сменилось 12 поколений процессоров *Intel* и *AMD*, различающихся микроархитектурой и технологией изготовления. Условная разбивка процессоров (а также компьютеров на их основе) по поколениям приведена в нижеследующей таблице.

Условная разбивка процессоров по поколениям

Поколение	Презентация первой модели процессора	Название характерной для поколения процессорной линейки	Микроархитектура	Технология изготовления (нм)
1-е	1978, июнь	<i>i8086 (i8088)</i>	-	3000
2-е	1982, февраль	<i>i80286</i>	-	1500
1-е	1982	<i>Am86</i>	-	3000
2-е	1985	<i>Am286</i>	-	1500
3-е	1985, октябрь	<i>i80386 (DX, SX)</i>	-	1500, 1000
4-е	1989, апрель	<i>i80486 (DX, DX2, DX4, SX, SX2)</i>	-	1000, 800, 600
3-е	1991, январь	<i>Am386 (DX, SX)</i>	-	1500, 1000
4-е	1993, январь	<i>Am486, Am5 × 86</i>	-	700, 500, 350
5-е	1993, март	<i>Pentium (P5, P54C, P54CS)</i>	-	800, 600, 350
6-е	1995 – 1999	<i>Pentium, Pro Pentium, II Pentium III</i>	<i>P6</i>	600, 350, 180
5-е	1996, январь	<i>K5</i>	<i>K5</i>	500, 350
6-е	1997 – 1999	<i>K6, K6-2, K6-III</i>	<i>K6</i>	350, 250
7-е	2000, ноябрь	<i>Pentium 4 до ядра Prescott</i>	<i>NetBurst</i>	180, 130
8-е	2005, май	<i>Pentium D (2-ядерные)</i>	<i>NetBurst</i>	90, 65
7-е	1999 – 2001	<i>Athlon, Athlon XP</i>	<i>K7</i>	250, 180
8-е	2003 – 2005	<i>Athlon 64, Athlon 64 X2 (2-ядерные)</i>	<i>K8</i>	130, 90

Окончание таблицы

Поколение	Презентация первой модели процессора	Название характерной для поколения процессорной линейки	Микроархитектура	Технология изготовления (нм)
9-е	2006, июль	<i>Core 2 (Duo, Quad – 2- и 4-ядерные)</i>	<i>Core</i>	65, 45
10-е	2008, ноябрь	<i>Core i7 (4- и 6-ядерные)</i>	<i>Nehalem</i>	45, 32
9-е	2007, ноябрь	<i>Phenom (4-ядерные)</i>	<i>K10</i>	65
10-е	2009, январь	<i>Phenom II (4-ядерные)</i>	«K10.5»	45
11-е	2011, январь	<i>Core i7 поколение 2 (4- и 6-ядерные)</i>	<i>Sandy Bridge</i>	32, 22
12-е	2013 – 2014	<i>Naswell, Broadwell</i>	<i>Broadwell</i>	22,14
11-е	2011, июнь	<i>Zambezi (4-, 6- и 8-ядерные)</i>	<i>Bulldozer</i>	32
12-е	2013, апрель	<i>Zambezi (8-ядерные)</i>	<i>Bulldozer</i>	28

Примечание. Серым цветом в таблице выделены строки моделей процессора AMD.

В каждом поколении имеются модификации, отличающиеся друг от друга назначением и ценой.

Тип исполнения процессора, его «внешность» и способ подключения к материнской плате определяет **форм-фактор**.

Форм-фактор процессора – это тип изготовления процессора, его конструктивного исполнения и способа подключения к материнской плате.

Процессоры квадратной формы (тип разъема сокет (*socket*)) имеют корпус с ножками-контактами. Существует прямоугольная форма слот (*slot*) – ножки-контакты размещены с одного бока процессорной платы.

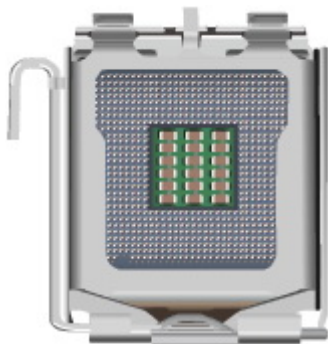


Рис. 6.1. Процессор Socket LGA 775

Например, старые модели процессоров имели форм-фактор *Socket7* (процессоры фирмы AMD-K6, K6-2, K6-2+, K6-3, Intel – Pentium II (200 – 266), Cyrix (M2), Centaur Technology (IDT)).

По форм-фактору *Slot1* выполнены процессоры фирмы Intel: Pentium II (233 – 600 МГц), Pentium III (*Katmai*) и Celeron (300 – 400 МГц).

Эра разъемов (сокетов) современных процессоров начинается с *Socket 775* или *Socket T* –

первого разъема под процессоры Intel, не имеющие гнезд (рис. 6.1). Разъем *Socket* выполнен в форм-факторе квадратной конфигурации с выступающими контактами. Процессор устанавливался на выступающие контакты, опускалась прижимная пластина, и с помощью рычажка он придавливался к контактам.

Разъем *Socket* до сих пор используется во многих персональных компьютерах. Предназначался для работы практически со всеми процессорами *Intel* четвертого поколения – *Pentium 4*, *Pentium 4 Extreme Edition*, *Celeron D*, *Pentium Dual-Core*, *Pentium D*, *Core 2 Quad*, *Core 2 Duo* и процессорами серии *Xeon*. Скоростные характеристики устанавливаемых процессоров – от 1400 до 3800 МГц.

Socket M – самый распространенный мобильный сокет. Использовался практически для всех мобильных процессоров компании *Intel*, до сих пор актуален в производстве ноутбуков. Выполнен в 478-м контактном исполнении. Предназначался для процессоров *Intel* – *Celeron*, *Core Solo*, *Core 2 Duo*, *Core Duo*, *Celeron M*. Выпускается с 2006 г. Скоростные характеристики процессоров – от 1600 до 3000 МГц.

Socket J, или сокет *LGA 771*, обновленный в 2006 г. серверный сокет, имеет исполнение с выступающими контактами. Предназначался для серверных решений. Сокет используют такие процессоры *Intel* – *Dual-Core* и *Quad-Core* серии *Xeon*, *Core 2 Extreme QX9775*. Скоростные характеристики – от 2 ГГц и выше.

Socket P – современный разъем для мобильных процессоров. Имеет 478 контактных гнезд. Выпускается с 2007 г. Предназначается под все мобильные процессоры компании *Intel* – *Dual-Core* с *T5xxx* по *T9xxx*, *Pentium Dual-Core* с *T23xx* по *T4xxx*, *Core 2 Quad*. Скоростные характеристики – от 1,6 ГГц и выше.

Socket 441 – специальный разъем, созданный для процессоров *Intel Atom*. Используется только для данных маломощных процессоров. Выпускается с 2008 г. Скоростные характеристики – от 600 до 2100 МГц.

Socket LGA 1155, или *Socket H2* – предназначен для замены сокета *LGA 1156*. Поддерживает самый современный процессор *Sandy Bridge* и будущий *Ivy Bridge*. Разъем выполнен в 1155-контактном исполнении. Выпускается с 2011 г. Скоростные характеристики – до 20 Гб/с.

Socket LGA 2011, или *Socket R* – последняя разработка компании *Intel*, заменяет *LGA 1366* (рис. 6.2). Разъем выполнен в 2011-контактном исполнении. Поддерживает процессор *Sandy Bridge*. Скоростные характеристики – от 19 до 25,6 Гб/с.



Рис. 6.2. Процессор *Socket R*

Основные параметры микропроцессоров:

- разрядность;
- рабочая тактовая частота;
- размер кэш-памяти;
- состав инструкций;
- конструктив;
- рабочее напряжение, энергопотребление, температурный режим и т. д.

Разрядность шины данных микропроцессора определяет количество разрядов, над которыми одновременно могут выполняться *операции*.

Разрядность шины адреса МП определяет его адресное пространство.

Адресное пространство – это максимальное количество ячеек основной памяти, которое может быть непосредственно адресовано микропроцессором.

Рабочая тактовая частота МП во многом определяет его внутреннее быстродействие, так как каждая команда выполняется за определенное количество тактов.

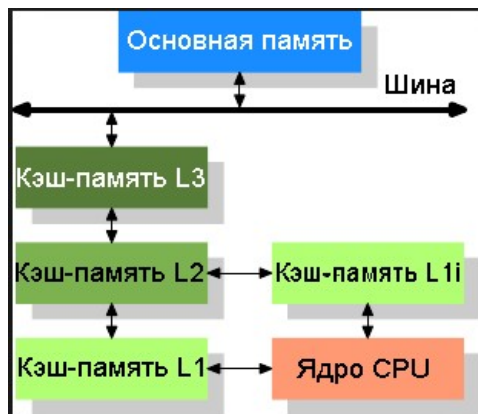


Рис. 6.3. Уровни кэш-памяти

Кэш-память имеет три уровня (рис. 6.3):

- *L1* – память 1-го уровня, находящаяся внутри основной микросхемы (ядра) МП и работающая всегда на полной частоте МП, может быть разделена на кэш данных *L1d-Cache* и кэш инструкций (команд) *L1i-Cache*;
- *L2* – память 2-го уровня, кристалл, размещаемый на плате МП или системной плате и связанный с ядром внутренней микропроцессорной шиной (впервые введен в МП *Pentium II*). Память *L2* может работать на полной или половинной частоте МП;
- *L3* – память 3-го уровня, имеется не во всех процессорах.

1-й уровень кэша – самая высокая скорость работы, но и самый маленький размер. В нем содержатся основные инструкции (алгоритмы), необходимые для работы процессора.

2-й уровень кэша – скорость работы чуть медленнее, но кэш имеет больший размер и работает с данными приложения. Здесь можно сказать, что 256 Кб на ядро считается слишком маленьким показателем, такой размер кэша 2-го уровня устанавливается только в самых дешевых процессорах. 512 Кб кэша 2-го уровня на ядро является нормальным для офисного компьютера. И 1 Мб кэша второго уровня достаточно для домашнего мультимедийного компьютера.

3-й уровень кэша менее скоростной по сравнению со вторым. Процессоры, позиционирующиеся как мощные мультимедийные, имеют порядка 3 – 6 Мб общего кэша 3-го уровня (для всех ядер). Топовые дорогие процессоры могут иметь 8 Мб и более общего кэша 3-го уровня.

Состав инструкций – перечень, вид и тип команд, автоматически исполняемых МП. Они определяют непосредственно те процедуры, которые могут выполняться над данными в МП.

Конструктив определяет те физические разъемные соединения, в которые устанавливается МП (*Slot* – щелевой (прямоугольный) разъем, *Socket* – разъем-гнездо (квадратный)).

Рабочее напряжение определяет пригодность материнской платы для установки МП.

Энергопотребление зависит от количества ядер, их частоты и технологического процесса. Энергопотребление современных процессоров колеблется в пределах 65 – 125 Вт, указывается на их упаковке и на сайте производителя.

Температурный режим приравнивается к максимальному энергопотреблению процессора и характеризуется таким показателем, как максимальный температурный пакет «*Thermal Design Power*», или «*TDP*». Для современных процессоров он также составляет 65 – 125 Вт. Здесь нужно учесть, что для процессора с *TDP* 65 Вт хватит самого простого и дешевого кулера (охладителя), с *TDP* 100 Вт кулер нужен помощнее, желательно с двумя-четырьмя тепловыми трубками, с *TDP* 125 Вт – кулер с четырьмя тепловыми трубками и более. На рис. 6.4 представлено фото кулера с тепловыми трубками.



Рис. 6.4. Кулер с тепловыми трубками

Все микропроцессоры можно разделить на четыре группы:

1. МП типа **CISC** (*Complex Instruction Set Command*) с полным набором системы команд.

2. МП типа **RISC** (*Redused Instruction Set Command*) с усеченным набором системы команд.

3. МП типа **VLIW** (*Very Length Instruction Word*) со сверхбольшим командным словом.

4. МП типа **MISC** (*Minimum Instruction Set Command*) с минимальным набором системы команд и весьма высоким быстродействием и т.д.

6.2. Микропроцессоры типа CISC

Основные особенности:

Имеют встроенный математический сопроцессор, могут работать с *умножением внутренней частоты*.

Характеризуются конвейерным выполнением команд. В них предусмотрены регистры для очереди команд. *Конвейерное выполнение команд* – это одновременное выполнение разных тактов последовательных команд в разных частях МП, что увеличивает эффективное быстродействие ПК в 2 – 5 раз;

Микропроцессоры работают в двух режимах:

- Реальном (однозадачном, *Real Address Mode*), в котором возможно выполнение только одной программы;
- Защищенном (многозадачном, *Protected Virtual Address Mode*), обеспечивающим выполнение сразу нескольких программ, а также непосредственную адресацию и прямой доступ к расширенной основной памяти.

Имеется поддержка системы виртуальных машин, когда каждая задача может выполняться под управлением своей операционной системы, то есть практически в одном МП моделируется как бы несколько компьютеров, работающих параллельно и имеющих разные операционные системы.

Все перечисленные особенности характерны для процессоров линии *Intel Pentium*, вплоть до 5-го поколения.

6.3. Микропроцессоры типа *RISC*

Микропроцессоры типа *RISC* содержат только набор простых, чаще всего встречающихся в программах команд. При необходимости выполнения более сложных команд в микропроцессоре производится их автоматическая сборка из простых.

В этих МП все простые команды имеют один размер и на выполнение каждой из них тратится один машинный такт (на выполнение даже самой короткой команды из системы *CISC* обычно тратится четыре такта). Простая архитектура позволяет удешевить процессор, поднять тактовую частоту, а также распараллелить исполнение команд между несколькими блоками исполнения (так называемые суперскалярные архитектуры процессоров).

Один из первых МП типа *RISC* – *ARM* (на его основе был создан ПК *IBM PC RT*). В настоящее время эту технологию используют многоядерные процессоры 6-го и выше поколения *Intel* (2, 4, 6, 8 ядер), выполненные по 65 нм (45 нм, 32 нм) техпроцессу.

Основные характеристики ЦП группы *RISC* (на примере процессора *Intel Core i5-661*) приведены на рис. 6.5.

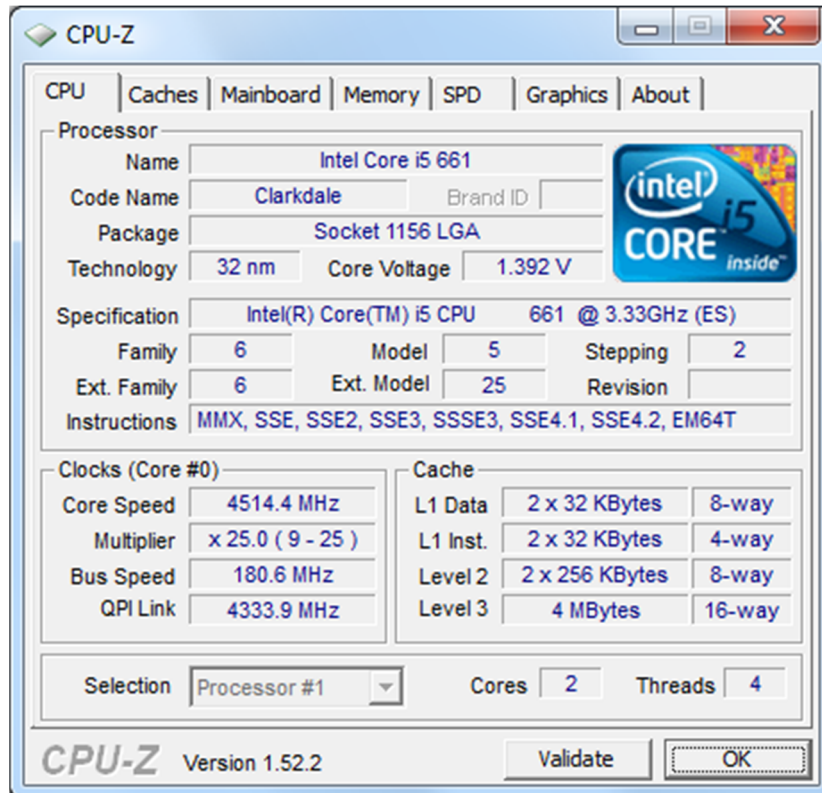


Рис. 6.5. Основные характеристики ЦП группы *RISC*

Современные 64-разрядные *RISC* МП выпускаются многими фирмами: *Apple (Power PC)*, *IBM (PPC)*, *HP (PA)* и т.д.

Микропроцессоры типа *RISC* имеют очень высокое быстродействие, но программно не совместимы с *CISC*-процессорами.

6.4. Микропроцессоры типа *VLIW* и *MISC*

Архитектуры *VLIW* (*Very Long Instruction Word* – Очень Длинное Слово Команды) отличаются от суперскалярной архитектуры тем, что решение о распараллеливании принимается не аппаратурой на этапе исполнения, а компилятором на этапе генерации кода. Команды очень длинные и содержат явные инструкции по распараллеливанию нескольких субкоманд на несколько устройств исполнения. Преимущество *VLIW* перед суперскалярной архитектурой – компилятор является более сложной и «умной» системой, способной хранить больше контекстной информации и принимать более верные решения об оптимизации.

MISC (англ. *Minimal Instruction Set Computer*) – процессор, работающий с минимальным набором длинных команд. Увеличение разрядности процессоров привело к идее укладки нескольких команд в одно большое слово. Это позволило использовать возросшую производительность компьютера и его возможность обрабатывать одновременно несколько потоков данных. Процессоры, образующие «компьютеры с минимальным набором команд» *MISC*, как и процессоры *RISC*, характеризуются небольшим числом чаще всего встречающихся команд. Вместе с этим принцип «очень длинных слов команд» *VLIW* обеспечивает выполнение группы непротиворечивых команд за один цикл работы процессора. Порядок выполнения команд распределяется таким образом, чтобы в максимальной степени загрузить маршруты, по которым проходят потоки данных. Таким образом архитектура *MISC* объединила вместе суперскалярную и *VLIW* концепции. Компоненты процессора просты и работают с высокими скоростями.

6.5. Физическая и функциональная структуры микропроцессора

Физическая структура микропроцессора достаточно сложна. На упрощенной структурной схеме одного ядра процессора *Intel* можно выделить две части (рис. 6.6):

- *операционную*, содержащую устройство управления (УУ), арифметико-логическое устройство (АЛУ), микропроцессорную память (за исключением нескольких адресных регистров) и регистр флагов (РФ);
- *интерфейсную*, содержащую адресные регистры; блок регистров команд – регистры памяти для хранения кодов команд, выполняемых в ближайшие такты работы машины; схемы управления шиной и портами.

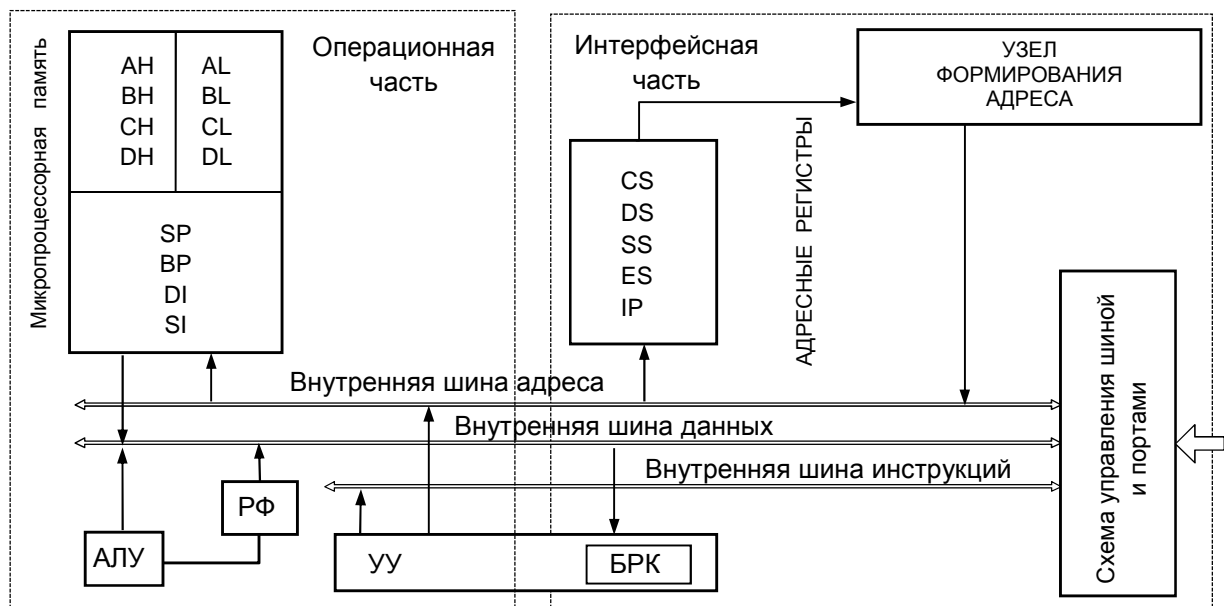


Рис. 6.6. Упрощенная структурная схема ЦП

Обе части МП работают параллельно, причем интерфейсная часть опережает операционную, так что выборка очередной команды из памяти (ее запись в блок регистров команд и предварительный анализ) выполняется во время выполнения операционной частью предыдущей команды.

6.6. Устройство управления

Устройство управления (УУ) является функционально наиболее сложным устройством ПК – оно вырабатывает управляющие сигналы, поступающие по кодовым шинам инструкций (КШИ) во все блоки машины. Упрощенная функциональная схема УУ показана на рис. 6.7.

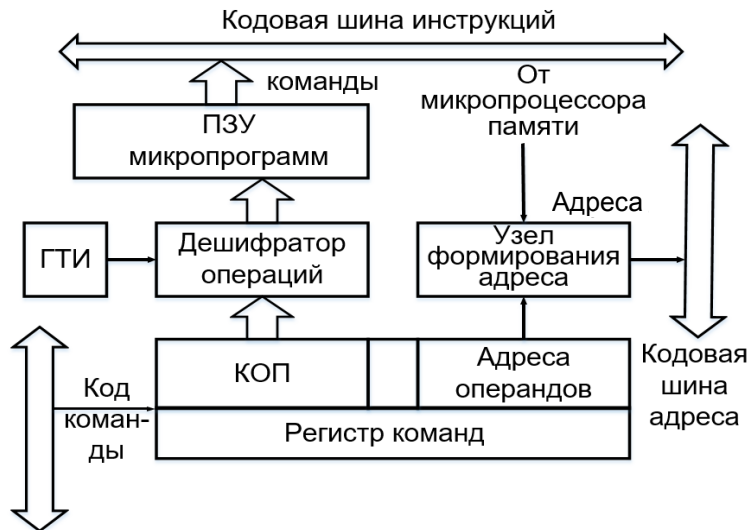


Рис. 6.7. Увеличенная функциональная схема УУ

На рисунке представлены:

- *регистр команд* – запоминающий регистр, в котором хранится код команды: код выполняемой операции (КОП) и адреса операндов, участвующих в операции; регистр команд расположен в интерфейсной части МП, в блоке регистров команд;
- *дешифратор операций* – логический блок, выбирающий в соответствии с поступающим из регистра команд КОП один из множества имеющихся у него выходов;
- *постоянное запоминающее устройство (ПЗУ) микропрограмм* хранит в своих ячейках управляющие сигналы (импульсы), необходимые для выполнения в блоках ПК процедур операций обработки информации;
- *узел формирования адреса* (находится в интерфейсной части МП) – устройство, вычисляющее полный адрес ячейки памяти (регистра) по реквизитам, поступающим из регистра команд и регистров МПП;
- *кодовые шины данных, адреса и инструкций* – часть внутренней интерфейсной шины микропроцессора.

6.7. Арифметико-логическое устройство

Арифметико-логическое устройство (АЛУ) предназначено для выполнения арифметических и логических операций преобразования информации. Функционально АЛУ (рис. 6.8) состоит обычно из двух регистров, сумматора и схем управления (местного устройства управления).

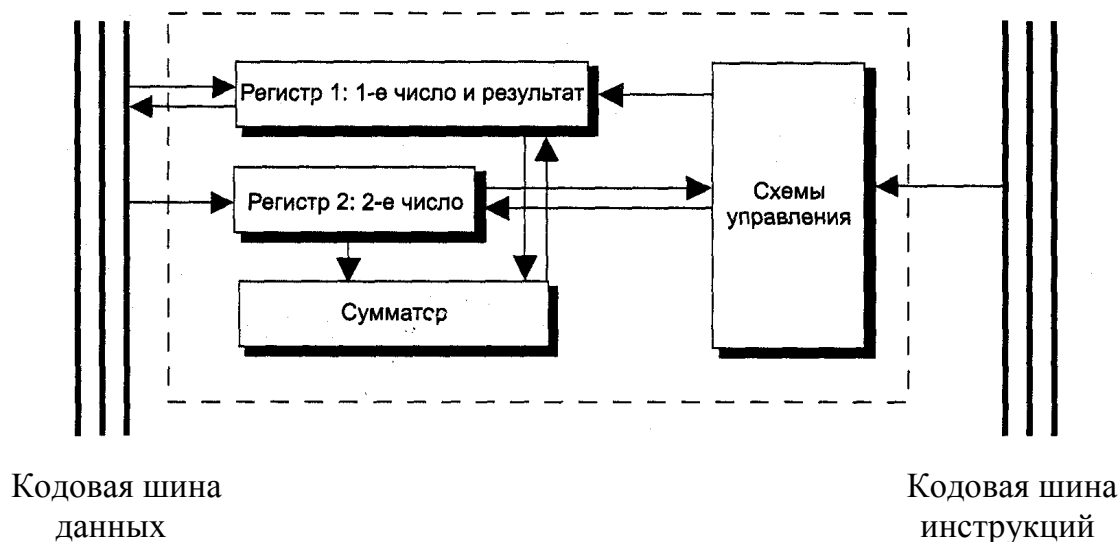


Рис. 6.8. Функциональная схема АЛУ

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 (Рг1) имеет разрядность двойного слова, а регистр 2 (Рг2) – разрядность слова.

При выполнении операций в Рг1 помещается первое число, участвующее в операции, а по завершении операции – результат; в Рг2 – 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 «+», «-», «x» и «÷» только над двоичной информацией с запятой, фиксированной после последнего разряда, т.е. только над целыми двоичными числами. Операции над двоичными числами с плавающей запятой и над двоично-кодированными десятичными числами выполняются с привлечением математического сопроцессора или по специально составленным программам.

6.8. Микропроцессорная память

Микропроцессорная память (МПП) базового МП 8088 включает в себя четырнадцать 2-байтовых запоминающих регистров. У МП 80386 и выше регистры 4-байтовые (у МП *Pentium* есть и 8-байтовые регистры).

Все регистры можно разделить на четыре группы:

- универсальные регистры: *AX, DX, CX, BX*;
- сегментные регистры: *CS, DS, SS, ES*;
- регистры смещений: *IP, SP, BP, SI, DI*;
- регистр флагов *FL*.

Универсальные регистры, или регистры общего назначения (РОН), используются для временного хранения любых данных, при этом можно работать с каждым регистром целиком, а можно отдельно и с каждой его половиной (регистры *AH, BH, CH, DH* – старшие (*High*) байты, а регистры *AL, BL, CL, DL* – младшие (*Low*) байты соответствующих 2-байтовых регистров). Но каждый из универсальных регистров может использоваться и как специальный при выполнении некоторых конкретных команд программы. В частности:

- регистр *AX* – регистр-аккумулятор, при выполнении операций умножения и деления *AX* используется для хранения первого числа, участвующего в операции, и результата операции;
- регистр *BX* часто используется для хранения адреса базы и начального адреса поля памяти при работе с массивами;
- регистр *CX* – регистр-счетчик, используется как счетчик числа повторений при циклических операциях;
- регистр *DX* используется как расширение регистра-аккумулятора при работе с 32-разрядными числами.

Сегментные регистры и регистры смещений

Сегментные регистры *CS, DS, SS, ES* используются для хранения начальных адресов полей памяти (сегментов), отведенных в программах для хранения:

- команд программы (сегмент кода – *CS*);
- данных (сегмент данных – *DS*);
- стековой области памяти (сегмент стека – *SS*);
- дополнительной области памяти данных при межсегментных пересылках.

Регистры смещений IP, SP, BP, SI, DI используются для хранения относительных адресов ячеек памяти внутри сегментов (смещений относительно начала сегментов).

Регистр флагов

Регистр флагов *FL* содержит условные одноразрядные признаки-маски, или флаги, управляющие прохождением программы в ПК; флаги работают независимо друг от друга и лишь для удобства помещены в единый регистр. Всего в регистре содержится девять флагов: шесть из них – *статусные*, отражают результаты операций, выполненных в компьютере (например, флаг нуля; устанавливается в 1, если результат операции равен 0; если результат не равен 0, то *ZF* обнуляется). Три других – *управляющие*, они непосредственно определяют режим исполнения программы, например управление прерываниями.

6.9. Интерфейсная часть ЦП

Интерфейсная часть МП предназначена для связи и согласования МП с системной шиной ПК, а также для приема, предварительного анализа команд выполняемой программы и формирования полных адресов операндов и команд. Интерфейсная часть включает в свой состав:

- адресные регистры МП;
- узел формирования адреса;
- блок регистров команд, являющийся буфером команд в МП;
- внутреннюю интерфейсную шину МП;
- схемы управления шиной и портами ввода-вывода.

Некоторые из названных устройств, такие как узел формирования адреса и регистр команды, непосредственно выполняемой МП, функционально входят в состав устройства управления.

6.10. Порты ввода-вывода

Это пункты системного интерфейса ПК, через которые МП обменивается информацией с другими устройствами. Каждый порт имеет адрес – номер порта; по существу, это адрес ячейки памяти, являющейся частью устройства ввода-вывода, использующего этот порт.

Схема управления шиной и портами выполняет следующие функции:

- формирование адреса порта;
- прием управляющей информации от порта;

- организация сквозного канала.

Схема управления шиной и портами использует для связи с портами кодовые шины инструкций, адреса и данные системной шины.

6.11. Функциональные характеристики ЦП

Тактовая частота обработки информации

Тактом называют интервал времени между началом подачи двух последовательных импульсов электрического тока, синхронизирующих работу различных устройств компьютера. Специальные импульсы для отсчета времени для всех электронных устройств вырабатывает тактовый генератор частоты, расположенный на системной плате. Его главный элемент представляет собой кристалл кварца, обладающий стабильностью резонансной частоты.

Тактовая частота определяется как количество тактов в секунду и измеряется в мегагерцах ($1 \text{ МГц} = 1 \text{ млн тактов/с}$). Тактовая частота влияет на *скорость работы, быстродействие* МП. Переход к микропроцессору с большей тактовой частотой означает скорое повышение обработки информации. Говоря о быстродействии процессора, имеют в виду количество операций, выполняемых им в секунду.

Для улучшения показателей при выполнении операций с плавающей запятой создано и используется специальное устройство – *математический сопроцессор*. Это интегральная схема, работающая во взаимодействии с центральным МП. Она предназначена только для выполнения математических операций.

Разрядность процессора

Это число одновременно обрабатываемых процессором битов, то есть количество внутренних битовых (двоичных) разрядов – важнейший фактор производительности МП. Процессор может быть 8-, 16-, 32- и 64-разрядным. Вместе с быстродействием разрядность характеризует объем информации, перерабатываемый процессором компьютера за единицу времени.

Интерфейс с системной шиной

Разрядность внутренней шины данных МП может не совпадать с количеством внешних выводов для линии данных. Например, МП с 32-разрядной внутренней шиной данных может иметь только 16 внешних линий данных. Это означает, что разрядность интерфейса с внешней

шиной данных равна 16. Аналогичная ситуация может наблюдаться с другой частью системной шины – адресной шиной. Как уже отмечалось, выполнение процессором команды предусматривает наряду с арифметическими действиями и логическими операциями передачу управления и перемещение данных из одного места памяти в другое. Поэтому важна не только разрядность внутренних шин процессора, но и его интерфейс с системной шиной.

Адресное пространство (адресация памяти)

Одна из функций процессора состоит в перемещении данных, в организации их обмена с внешними устройствами и оперативной памятью. При этом процессор формирует код устройства, а для ОЗУ – адрес ячейки памяти. Код адреса передается по адресной шине. Объем физически адресуемой микропроцессором оперативной памяти называется его *адресным пространством*. Он определяется разрядностью внешней шины адреса.

Действительно, пусть N – разрядность адресной шины, тогда количество различных двоичных чисел, которые можно по ней передать, равно 2^N . Известно, что число, передаваемое по адресной шине при обращении процессора к оперативной памяти, есть адрес ячейки ОЗУ (ее порядковый номер). Значит, 2^N – это количество ячеек оперативной памяти, к которым, используя адресную шину, может обратиться (адресоваться) процессор, то есть 2^N – объем адресного пространства процессора. Следовательно, при 16-, 20-, 24- или 32-разрядной шине адреса создается адресное пространство соответственно $2^{16} = 64$ Кбайта, $2^{20} = 1$ Мбайт, $2^{24} = 16$ Мбайт, $2^{32} = 4$ Гбайта.

6.12. Система команд ЦП

Семейство микропроцессоров фирмы *Intel*, начиная от 8086, имеет базовую систему команд, в состав которой входят следующие группы:

- команды пересылки данных:
 - команды пересылки данных внутри МП (*MOV, PUSH, POP, XCHG* и т.д.);
 - команды ввода-вывода (*IN, OUT*);
 - операции с флагами;
 - операции с адресами (*LEA, LDS* и т.д.);

- арифметические команды:
 - основные (сложение, вычитание, умножение, деление);
 - дополнительные (*INS*, *DEC* и др.);
- логические команды (сдвиг, дизъюнкция, конъюнкция, отрицание равнозначности и др.);
- команды обработки строковых данных (пересылка, сравнение, сканирование, слияние/разделение и др.);
- команды передачи управления (безусловный переход, условный переход, прерывания, переход с возвратом);
- команды управления («нет операции», «внешняя синхронизация» и т.д.). Каждая команда имеет большое число модификаций, чаще всего определяемых режимом адресации данных (операндов).

6.13. Взаимодействие элементов при работе ЦП

Работой МП управляет программа, записанная в ОП ЭВМ. Адрес очередной команды хранится в счетчике команд *IP* (*Instruction Pointer*) и в одном из сегментных регистров, чаще всего в *CS*. Каждый из них в реальном режиме имеет длину 16 бит, тогда как физический адрес оперативной памяти (ОП) должен иметь длину 20 бит. Несогласованность длины машинного слова (16 бит) и длины физического адреса ОП (20 бит) приводит к тому, что в командах невозможно указать физический адрес ОП – его приходится формировать, собирать из разных регистров МП в процессе работы.

В реальном режиме вся ОП делится на сегменты (длина сегмента – 64 Кб). Адрес ОП разделяется на две части: номер сегмента в ОП (база сегмента) и номер ячейки внутри данного сегмента (смещение относительно начала сегмента). Базовый адрес сегмента образуется добавлением к номеру сегмента справа четырех нулей. Поскольку последние четыре разряда абсолютного (физического) адреса сегмента всегда нулевые, сегмент может начинаться не с любой ячейки ОП, а только с «параграфа» – начала 16-байтного блока ОП. В структуре микропроцессора (как уже известно) имеется несколько регистров сегментов, например в *i8086* – четыре:

CS – программный сегмент;

DS – сегмент данных (информационный сегмент);

SS – стековый сегмент;

ES – расширенный сегмент (дополнительный сегмент данных).

Номер ячейки внутри сегмента (смещение) называется также *исполнительным адресом*. В большинстве случаев в адресной части команды указывается именно исполнительный адрес – номер сегмента чаще всего подразумевается по умолчанию. Однако допускается указание и полного адреса ОП в виде префиксной структуры: «сегмент: смещение». Если сегмент в команде не указывается, значит работа ведется внутри текущего сегмента (характер выполняемой работы и какой из сегментных регистров определяет текущую базу сегмента зависят от вида выполняемой команды).

Номер сегмента так же, как и смещение, имеет длину 2 байта. При вычислении физического адреса ОП сегмент и смещение суммируются, но сегмент перед суммированием сдвигается влево на 4 бита. В результате суммирования образуется физический адрес ОП длиной 20 бит.

В защищенном режиме базовые адреса сегментов хранятся в *дескрипторных* таблицах и имеют длину 24 или 32 бита (в зависимости от типа МП). В сегментных же регистрах хранится селектор, содержащий номер дескрипторной таблицы и дескрипторное смещение, т.е. порядковый номер дескриптора (в котором и хранится базовый адрес сегмента) в данной дескрипторной таблице (рис. 6.9).

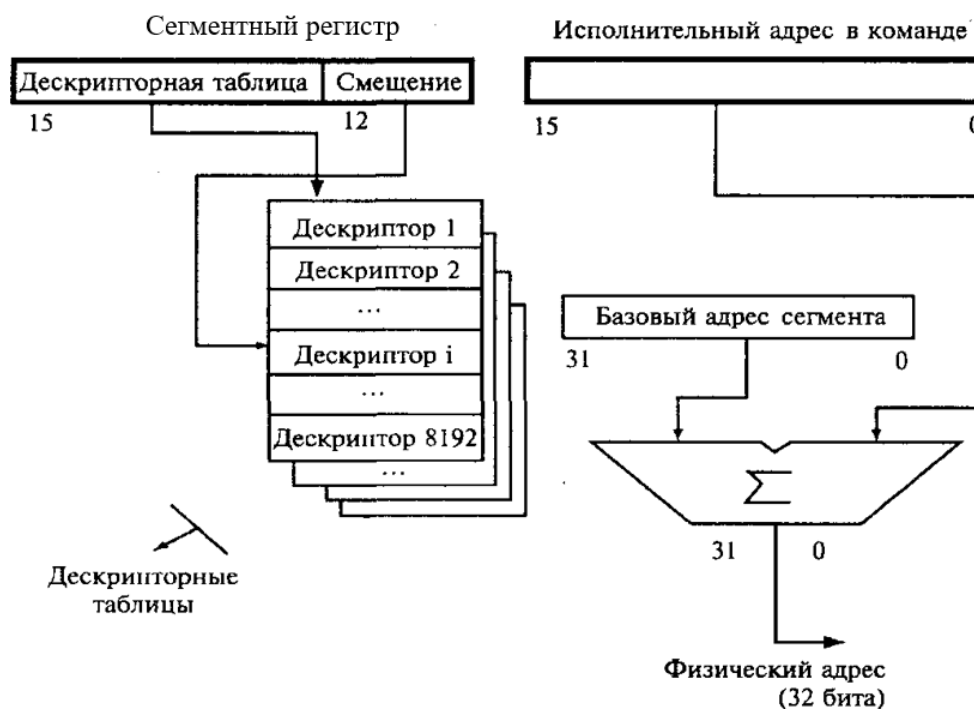


Рис. 6.9. Формирование физического адреса ОП в защищенном режиме

Физический адрес очередной команды через внутреннюю магистраль МП и интерфейс памяти поступает на шину адреса системной магистрали. Одновременно из устройства управления (УУ) исполнительного блока на шину управления выдается команда (управляющий сигнал) в ОП, предписывающая выбрать число, находящееся по адресу, указанному в системной магистрали. Выбранное число, являющееся очередной командой, поступает из ОП через шину данных системной магистрали, интерфейс памяти, внутреннюю магистраль МП на регистр команд (*INST*).

Из команды в регистре команд выделяется код операции, который поступает в УУ исполнительного блока для выработки управляющих сигналов, настраивающих микропроцессор на выполнение требуемой операции.

В зависимости от используемого в команде режима адресации организуется выборка необходимых исходных данных.

6.14. Работа центрального процессора при выполнении задания пользователя

Управляющая работой ЭВМ программа перед началом выполнения (например, с винчестера) загружается в основную память (рис. 6.10). Адрес первой выполняемой команды передается процессору и запоминается в счетчике команд (*IP*).

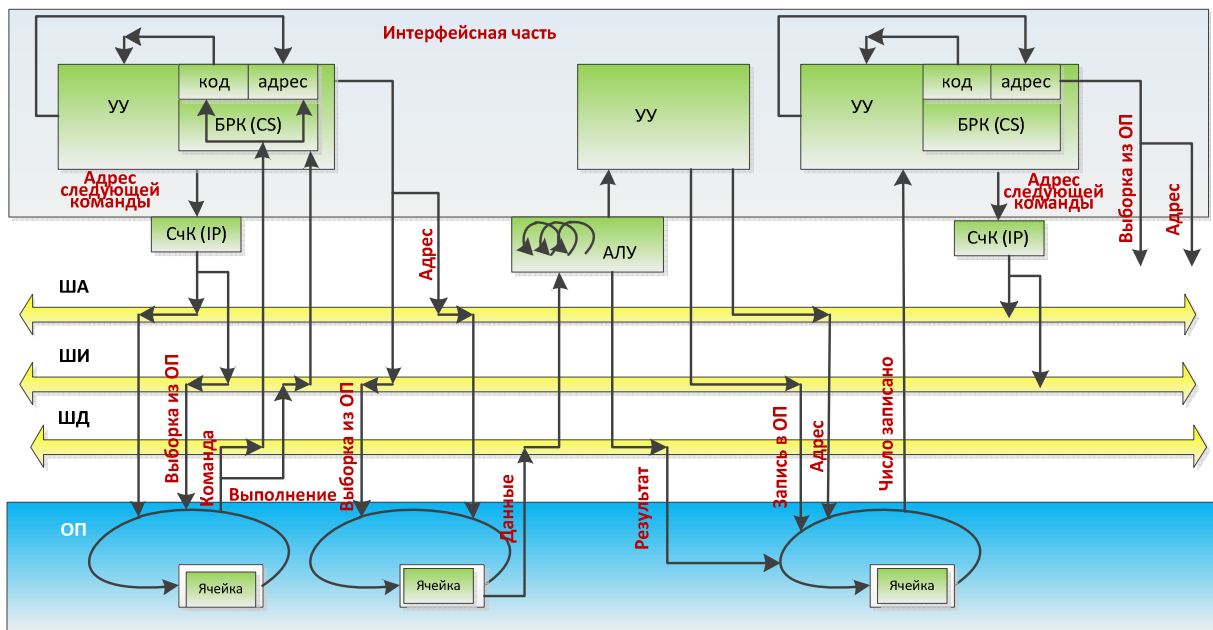


Рис. 6.10. Схема работы центрального процессора при выполнении задания пользователя

Начало работы процессора заключается в том, что адрес из счетчика команд (в котором всегда хранится адрес очередной команды) выставляется на шину адреса (ША) системной магистрали. Одновременно на шину инструкций (ШИ) выдается команда «*выборка из ОП*», которая воспринимается основной памятью. Получив с шины инструкций системной магистрали команду, основная память считывает адрес с шины адреса, находит ячейку с этим номером и ее содержимое выставляет на шину данных (ШД), а на шину инструкций выставляет сигнал о выполнении команды.

Процессор, получив по шине инструкций сигнал об окончании работы ОП, вводит число с шины данных на внутреннюю магистраль МП и через нее пересылает введенную информацию в регистр команд.

В регистре команд полученная команда разделяется на кодовую и адресную части. Код команды поступает в блок управления для выработки сигналов, настраивающих МП на выполнение заданной операции, и для определения адреса следующей команды (который сразу заносится в счетчик команд). Адресная часть команды выставляется на шину адреса системной магистрали (СМ) и сопровождается сигналом «*выборка из ОП*» данных на шине инструкций. Выбранная из ОП информация через шину данных поступает на внутреннюю магистраль МП, с которой вводится в арифметико-логическое устройство. На этом заканчивается подготовка МП к выполнению операции и начинается ее выполнение в АЛУ.

Результат выполнения операции выставляется микропроцессором на шину данных, на шину адреса выставляется адрес ОП, по которому этот результат необходимо записать, а на шину инструкций выставляется команда «*запись в ОП*». Получив с шины инструкций команду, ОП считывает адрес и данные с системной магистрали, организует запись данных по указанному адресу и после выполнения команды выставляет на шину управления сигнал, обозначающий, что число записано. Процессор, получив этот сигнал, начинает выборку очередной команды: выставляет адрес из счетчика команд на шину адреса, формирует команду «*выборка из ОП*» на шине инструкций и т.д.

В каждом цикле, получив команду в регистр команд и выделив код операции, процессор определяет, к какому устройству она относится. Если команда должна выполняться процессором, организуется ее выполнение по описанному циклу. Если же команда предназначена для выполнения в другом устройстве ЭВМ, центральный процессор (ЦП) передает ее соответствующему устройству.

Процесс передачи команды другому устройству

В этом случае выполняются следующие действия: ЦП выставляет на шину адреса СМ адрес интересующего его устройства.

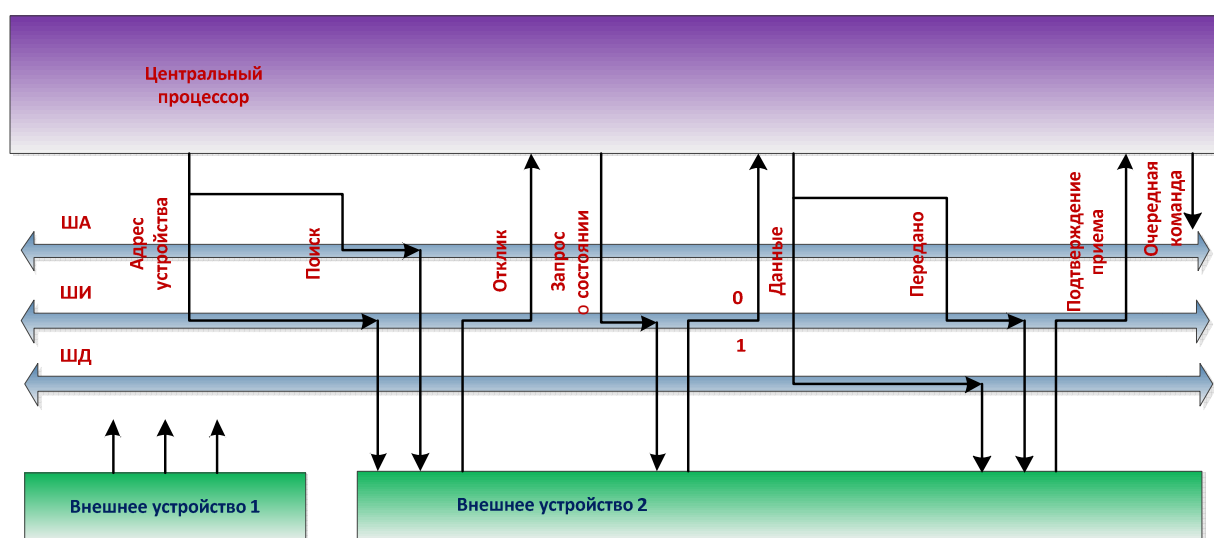


Рис. 6.11. Схема работы центрального процессора в процессе передачи команды другому устройству

По шинам инструкций передается сигнал «поиск устройства». Все устройства, подключенные к системной магистрали, получив этот сигнал, читают номер устройства с шины адреса и сравнивают его со своим номером. Устройства, для которых эти номера не совпадают, на эту команду не реагируют. Устройство с совпавшим номером вырабатывает сигнал отклика по шине инструкций. ЦП, получив сигнал отклика, в простейшем случае выставляет имеющуюся у него команду на шину данных и сопровождает ее по шине инструкций сигналом «передать команду». Получив сигнал о приеме команды, ЦП переходит к выполнению очередной своей команды, выставляя на шину адреса содержимое счетчика команд.

В более сложных случаях, получив сигнал, что устройство откликнулось, прежде чем передавать команду, ЦП запрашивает устройство о его состоянии. Текущее состояние устройства закодировано в байте состояния, который откликнувшееся устройство передает процессору через ШД системной магистрали. Если устройство включено и готово к работе, то байт состояния – нулевой. Наличие в нем единиц свидетельствует о нештатной ситуации, которую ЦП пытается проанализировать и в необходимых случаях извещает оператора о сложившейся ситуации.

Таким образом, взаимодействие МП с внешними устройствами предусматривает выполнение логической последовательности действий, связанных с поиском устройства, определением его технического состояния, обменом командами и информацией.

Режим, когда ЦП, запустив внешнее устройство, переходит в состояние ожидания и находится в нем до тех пор, пока внешнее устройство не сообщит ему об окончании обмена данными, называют *однопрограммным*. Это приводит к простоя большинства устройств ЭВМ, так как в каждый момент времени может работать только одно из них.

Для ликвидации таких простоев и повышения эффективности работы оборудования внешние устройства сделаны автономными. Получив от ЦП необходимую информацию, они самостоятельно организуют свою работу по обмену данными. Процессор же, запустив внешнее устройство, продолжает выполнение программы. При необходимости он может запустить в работу несколько других устройств или выполнение еще одной программы. Такой режим работы ЭВМ называется *многопрограммным*.

6.15. Работа микропроцессора при выполнении программного прерывания

Во время работы ЭВМ в ЦП поступает (и вырабатывается в нем самом) большое количество различных сигналов. Сигналы, которые выполняемая в ЦП программа способна воспринять, обработать и учесть, составляют *поле зрения ЦП* или, другими словами, входят в зону его внимания.

Например, если процессором исполняется программа сложения двух двойных слов, то в «поле ее зрения» находятся флаги микропроцессора, определяющие знаки исходных данных и результата, наличие переноса из тетрады или байта, переполнение разрядной сетки и др. Такая программа готова реагировать на любой из сигналов, находящихся в ее «зоне внимания» (а поскольку именно программа управляет работой ЦП, то она определяет и «зону внимания» центрального процессора). Но если во время выполнения такой программы нажать какую-либо клавишу, то эта программа «не заметит» сигнала от этой клавиши, так как он не входит в ее «поле зрения».

Для того чтобы ЦП, выполняя свою работу, имел возможность реагировать на события, происходящие вне его «зоны внимания», наступления которых он «не ожидает», существует *система прерываний ЭВМ*. Эта система позволяет организовать работу по диагностике и автоматическому восстановлению в момент возникновения нештатной ситуации, прервав выполнение основной работы таким образом, чтобы сохранить полученные к этому времени правильные результаты.

Система прерываний позволяет микропроцессору выполнять основную работу, не отвлекаясь на проверку состояния сложных систем при отсутствии такой необходимости, или прервать выполняемую работу и переключиться на анализ возникшей ситуации сразу после ее появления.

В зависимости от места нахождения источника прерываний они могут быть разделены на *внутренние (программные и аппаратные)* и *внешние* прерывания (поступающие в ЭВМ от внешних источников, например, от клавиатуры или модема).

Принцип действия системы прерываний заключается в следующем: при выполнении программы после каждого рабочего такта микропроцессора изменяются содержимое регистров, счетчиков, состояние отдельных управляющих триггеров, т.е. изменяется состояние процессора. Информация о состоянии процессора лежит в основе многих процедур управления вычислительным процессом. Не вся информация одинаково актуальна, есть существенные элементы, без которых невозможно продолжение работы. Эта информация должна сохраняться при каждом «переключении внимания процессора».

Совокупность значений наиболее существенных информационных элементов (содержимое счетчика команд, сегментных регистров, регистра флагов и др.) называется *вектором состояния*, или *словом состояния процессора* (в некоторых случаях она называется *словом состояния программы*).

При возникновении события, требующего немедленной реакции со стороны машины, ЦП прекращает обработку текущей программы и переходит к выполнению другой программы, специально предназначенной для данного события, по завершении которой возвращается к выполнению отложенной программы. Такой режим работы называется *прерыванием*.

Каждое событие, требующее прерывания, сопровождается специальным сигналом, который называется *запросом прерывания*. Программа, затребованная запросом прерывания, называется *обработчиком прерывания*.

Запросы на прерывание могут возникать из-за сбоев в аппаратуре (зафиксированных схемами контроля), переполнения разрядной сетки, деления на нуль, выхода за установленные для данной программы области памяти, затребования периферийным устройством операции ввода-вывода или завершения ее и т.д.

Прерывания делятся на три типа: *аппаратные, логические и программные*.

Аппаратные прерывания вырабатываются устройствами, требующими внимания микропроцессора: прерывание № 2 – отказ питания; № 8 – от таймера; № 9 – от клавиатуры и др.

Запросы на *логические* прерывания вырабатываются внутри микропроцессора при появлении «нештатных» ситуаций: прерывание № 0 – при попытке деления на 0; № 4 – при переполнении разрядной сетки арифметико-логического устройства; № 1 – при переводе микропроцессора в пошаговый режим работы и т.д. Запрос на *программное* прерывание формируется по команде *INT n*, где *n* – номер вызываемого прерывания. Запрос на аппаратное или логическое прерывание вырабатывается в виде специального электрического сигнала.

Вопросы к компьютерному тестированию

1. Какие из операций предусматривают выполнение процессором команды программы?
2. Как по-другому называют тип исполнения процессора, определяющий его «внешность» и способ подключения к материнской плате?
3. К какому типу относят процессоры квадратной формы, имеющие корпус с ножками-контактами?
4. К какому уровню относят кэш-память, размещаемую на плате МП или системной плате и связанную с ядром внутренней микропроцессорной шиной?
5. В какой из технологий процессоров на выполнение даже самой короткой команды обычно тратится четыре такта?
6. К какому типу относят процессоры с минимальным набором системы команд и высоким быстродействием?
7. Какую технологию используют современные многоядерные процессоры 6-го и выше поколения *Intel* (2, 4, 6, 8 ядер), выполненные по 65 нм (45 нм, 32 нм) техпроцессу?
8. В какой из технологий процессоров решение о распараллеливании принимается не аппаратурой на этапе исполнения, а компилятором на этапе генерации кода?
9. Какие узлы включает операционная часть физической структуры микропроцессора?
10. Какие узлы включает интерфейсная часть физической структуры микропроцессора?
11. Какие узлы включает устройство управления физической структуры микропроцессора?
12. Какие узлы включает арифметико-логическое устройство физической структуры микропроцессора?
13. С какой двоичной информацией может выполнять арифметические операции АЛУ?
14. Какие группы регистров включает в себя микропроцессорная память?
15. Как называют регистры микропроцессорной памяти, которые используются для хранения относительных адресов ячеек памяти внутри сегментов?

16. Какие из регистров микропроцессорной памяти используются для хранения начальных адресов полей памяти, отведенных в программах для хранения команд программы, данных и пр.?
17. Как называют часть упрощенной структурной схемы одного ядра процессора, предназначенную для связи и согласования МП с системной шиной ПК, а также для приема и предварительного анализа команд выполняемой программы?
18. Как называют пункты системного интерфейса ПК, через которые МП обменивается информацией с другими устройствами?
19. Какая из функциональных характеристик ЦП определяет скорость работы и быстродействие компьютера?
20. Какое устройство используется совместно с АЛУ для выполнения операций с плавающей запятой?
21. Как называют объем физически адресуемой микропроцессором оперативной памяти?
22. Какой объем адресного пространства процессора в килобайтах (гигабайтах) при 16 или 32-разрядной шине адреса?
23. К какой из групп системы команд центрального процессора относятся операции с флагами?
24. К какой из групп системы команд центрального процессора относятся операции прерывания?
25. Записать базовый адрес сегмента ОП, если его номер определяется двоичным числом 00001111.
26. В каком из регистров при работе ЦП запоминается адрес первой выполняемой команды?
27. Какая команда первой выставляется на шину инструкций в начале работы центрального процессора?
28. Как называют режим, когда внешние устройства самостоятельно организуют свою работу по обмену данными?
29. Как называют прерывания, вырабатываемые устройствами, требующие внимания микропроцессора?
30. Как называют прерывания, вырабатываемые внутри микропроцессора?

Глава 7

АДАПТЕРЫ. ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЭВМ



Иллюстрация

Рассматриваемые вопросы:

7.1. Адаптеры.

7.2. Запоминающие устройства ЭВМ.

7.2.1. Состав, устройство и принцип действия основных запоминающих устройств.

7.2.2. Внешние запоминающие устройства.

7.1. Адаптеры

Ввиду того что формы представления данных и управляющих сигналов в разных устройствах ПК существенно различаются, то для поддержки взаимодействия устройств необходимо выполнять соответствующие преобразования. Эту задачу решают специальные устройства, называемые *адаптерами*.

Конструктивно они оформляются в виде модулей, которые, с одной стороны, имеют стандартный разъем для сопряжения с шиной, а с другой – разъем (или разъемы) для связи с соответствующим устройством (рис. 7.1). На плате модуля размещаются микросхемы и другие элементы, которые и выполняют необходимые преобразования.



Рис. 7.1. Адаптер

Следует отметить тенденцию уменьшения потребности в адаптерах, так как часть «обязанностей» по преобразованию сигналов берут на себя

электронные схемы управления самих устройств (к примеру, тех же

накопителей), а некоторые согласования выполняют чипсеты. Тем не менее сегодня в номенклатуре адаптеров устойчиво фигурируют: видеоадаптеры (они же – видеоплаты, видеокарты), адаптеры портов ввода-вывода, сетевые адаптеры (сетевые карты), звуковые платы (аудиокарты), модемы.

Видеоадаптер – это устройство, преобразующее набор данных, подлежащих отображению на экране, в видеосигнал, посылаемый монитору по кабелю.

Контроллер дисков предназначен для управления работой механических подвижных частей устройства и формирования электрических импульсов при записи и чтении. Он содержит:

- генератор, питающий переменным током двигатель дисков;
- сложную сервосистему, которая управляет устройством позиционирования блока головок на требуемую дорожку (цилиндр) в соответствии с поступающими от адаптера сигналами;
- усилители записи, формирующие электрические импульсы, которые подаются на магнитные головки при записи данных;
- усилители считывания и формирователи выходных сигналов при считывании информации.

Контроллер ввода-вывода, или *адаптер портов*, представляет собой устройство, которое обслуживает разнообразные внешние устройства, такие как принтеры, манипуляторы и другие, присоединяющиеся к процессорному блоку через специальные схемные элементы, называемые *портами*.

Различают *параллельные* и *последовательные* порты. Параллельный порт позволяет передать за один такт по крайней мере один байт, поскольку каждому биту выделен один проводник (один контакт), и таким образом все составляющие байта передаются одновременно, параллельно.

Последовательный порт содержит для передачи данных только одну пару проводников, и потому биты, составляющие сигнал, проходят через порт последовательно.

Наиболее часто адаптер ввода-вывода обслуживает три параллельных порта (их именуют *LPT1 ... LPT3*) и четыре последовательных (с именами *COM1 ... COM4*). Для *LPT*-портов используют 41-штырьковые разъемы, для *COM*-портов – 9- или 25-штырьковые. Разъемы выхо-

дят на заднюю стенку системного блока, к ним подключаются соединительные кабели внешних устройств. Общее число разъемов, как правило, меньше числа портов.



Рис. 7.2. Сетевая плата

Сетевая плата. Этот адаптер предназначен для сопряжения персонального компьютера с физическим каналом передачи данных, например с коаксиальным кабелем (рис. 7.2). Он осуществляет двунаправленную транспортировку данных: прием сигналов из канала и передачу их на шину компьютера или, наоборот, – прием данных из компьютера и их передачу в канал. При этом сетевая плата выполняет все необходимые преобразования структуры передаваемых сообщений строго в соответствии со стандартами, по которым построена данная вычислительная сеть.

редаваемых сообщений строго в соответствии со стандартами, по которым построена данная вычислительная сеть.

7.2. Запоминающие устройства ЭВМ

Комплекс технических средств, реализующих функцию памяти, называется *запоминающим устройством* (ЗУ). Основное назначение ЗУ – размещение в них команд и данных. Они обеспечивают центральному процессору доступ к программам и информации.

Устройства памяти ЭВМ делят на основные и внешние запоминающие устройства.

7.2.1. Состав, устройство и принцип действия основных запоминающих устройств

Основные ЗУ подразделяют на *основную память* (ОП) и *сверхоперативную память* (СОЗУ).

Основная память

Основная память включает в себя два типа устройств: оперативное запоминающее устройство (ОЗУ, или от англ. *RAM – Random Access Memory*) и постоянное запоминающее устройство (ПЗУ, или от англ. *ROM – Read Only Memory*).

Оперативное запоминающее устройство

ОЗУ предназначено для хранения переменной информации, допускает изменение своего содержимого в ходе выполнения процессором вычислительных операций с данными и может работать в режимах

записи, чтения, хранения. Оперативная память – полигон, на котором компьютер проводит все свои операции. И, конечно же, чем шире этот полигон, тем лучше.

Основная составная часть микросхемы ОЗУ – массив элементов памяти (ЭП), объединенных в матрицу накопителя.

Каждый элемент памяти может хранить 1 бит информации и имеет свой адрес. ЗУ, позволяющие обращаться по адресу к любому ЭП в произвольном порядке, называются *запоминающими устройствами с произвольным доступом*.

При матричной организации памяти реализуется координатный принцип адресации ЭП, в связи с чем адрес делится на две части – две координаты X и Y . На пересечении этих координат находится элемент памяти, чья информация должна быть прочитана или изменена. ОЗУ связано с остальным микропроцессорным комплектом ЭВМ через системную магистраль (рис. 7.3).

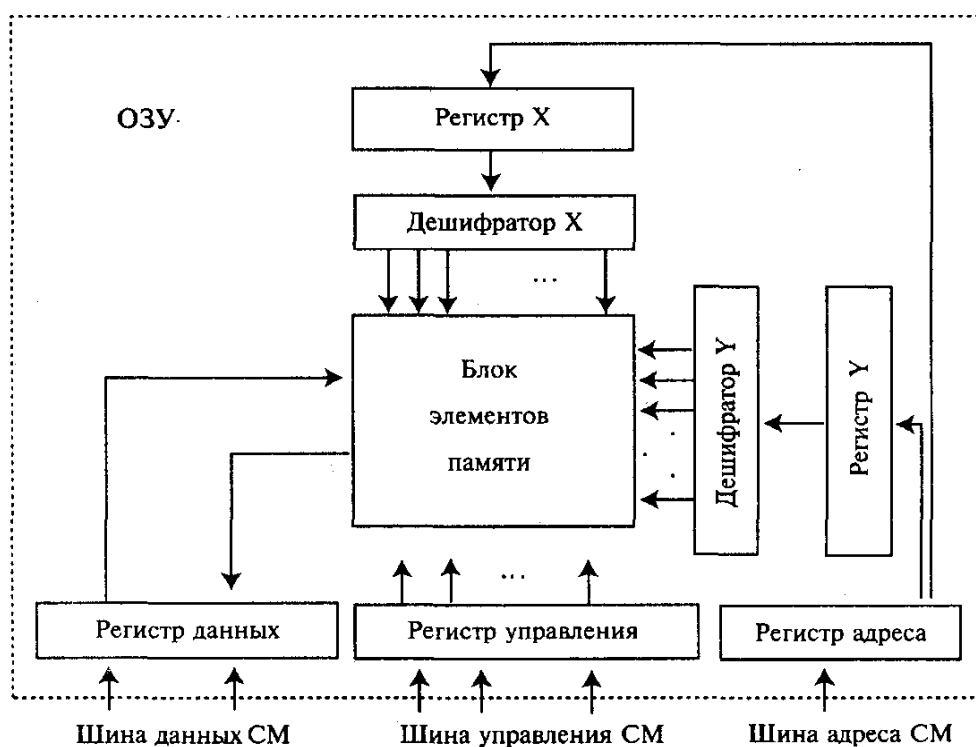


Рис. 7.3. Структурная схема ОЗУ

По шине управления передается сигнал, определяющий, какую операцию необходимо выполнить.

По шине данных передается информация, записываемая в память или считываемая из нее.

По шине адреса передается адрес участвующих в обмене элементов памяти. Максимальная емкость памяти определяется количеством линий в шине адреса системной магистрали: если количество линий обозначить m , то емкость памяти (т.е. количество элементов памяти, имеющих уникальные адреса) определяется как 2^m . Так, в *IBM PC XT* шина адреса системной магистрали (СМ) содержит 20 линий. Поэтому максимальный объем ОП в этих машинах равен $2^{20} = 1$ Мб. В *IBM PC AT* (с микропроцессором *i80286*) СМ содержит 24 линии, поэтому объем ОП может быть увеличен до 16 Мб. Начиная с МП *i80386*, шина адреса содержит 32 линии. Максимальный объем ОП увеличился до $2^{32} = 4$ Гб и более. У современных процессоров разрядность адресной шины достигает 64.

Микросхемы памяти могут строиться на *статических (SRAM)* и *динамических (DRAM)* ЭП. В качестве статического ЭП чаще всего выступает статический триггер. В качестве динамического ЭП может использоваться электрический конденсатор, сформированный внутри кремниевого кристалла.

Статические ЭП способны сохранять свое состояние (0 или 1) неограниченно долго (при включенном питании). Динамические ЭП с течением времени записанную в них информацию теряют (например, из-за саморазряда конденсатора), поэтому они нуждаются в периодическом восстановлении записанной в них информации – в регенерации.

Динамические ОЗУ отличаются меньшим числом компонентов в одном элементе памяти, в связи с чем имеют меньшие размеры и могут быть более плотно упакованы в кристалле. Однако из-за необходимости регенерации информации динамические ОЗУ имеют более сложные схемы управления.

Основные характеристики ОЗУ – объем и быстродействие.

В современных ПЭВМ ОЗУ имеет модульную структуру. Конструктивное исполнение модуля памяти называется его *форм-фактором*.

Выделяют ряд форм-факторов (устаревшие – *SIP*, *ZIP* и действующие – *SIMM*, *DIMM*, *FB-DIMM*, *SODIMM*, *MicroDIMM*, *RIMM*), которые совершенно не совместимы друг с другом:

SIMM (Single in Line Memory Module): имеют 30 или 72 контакта, каждый из которых выходит на обе стороны платы памяти (рис. 7.4). Последние могут использоваться в компьютерах с процессорами *Pentium* только парами.

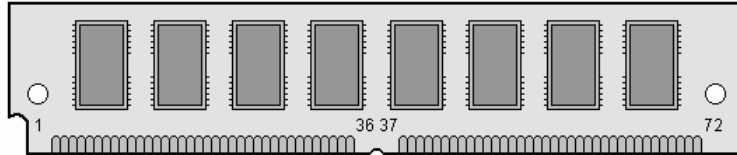


Рис. 7.4. Конструкция модуля *SIMM*

DIMM (Dual in Line Memory Module): имеют 168, 184, 200, 240 контактов с обеих сторон платы памяти (рис. 7.5).

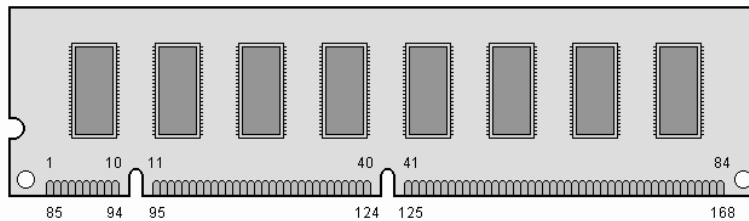


Рис. 7.5. Конструкция модуля *DIMM*

Модули памяти стандарта *FB-DIMM* предназначены для серверов. Схожи с модулями памяти *DIMM 240-pin*, но несовместимы с обычными небуферизованными модулями памяти *DDR2 DIMM* и *Registered DDR2 DIMM*.

SODIMM (Small Outline DIMM): имеют 72, 144, 168, 200 контактов. Подходит для ноутбуков и *Tablet PC*.

MicroDIMM: имеют 60 контактов. Используются в ноутбуках и субноутбуках.

RIMM: имеют 184, 168, 242 контакта.

Внешне модули *RIMM* напоминают модули *DIMM* (рис. 7.6, а), однако, во-первых, имеют меньшее число контактов, а во-вторых, с обеих сторон закрыты металлическим экраном (рис. 7.6, б), защищающим от наводок и взаимного влияния модулей, работающих на больших частотах (рис. 7.6).

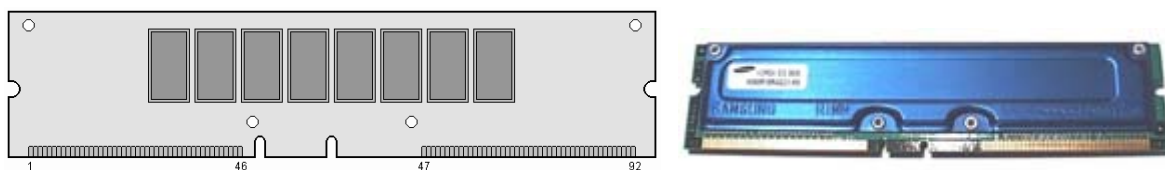


Рис. 7.6. Конструкция модуля *RIMM*

Типы оперативной памяти

- *SRAM* – статическая и довольно дорогая память давно устарела в качестве оперативной, потом устарела в качестве кэша. В *SRAM* ячейка памяти – это триггер из нескольких конденсаторов, которые удерживают заряд.
- *DRAM (Dynamic RAM)* – динамическая память. Отличается от *SRAM* типами ячеек. Заряд в конденсаторах памяти сохраняется очень непродолжительное время (приблизительно 2 мс), поэтому его нужно постоянно регенерировать.
- *SDRAM (Synchronous Dynamic Random Access Memory)* – синхронная динамическая память со случайным доступом. По сравнению с памятью предыдущих поколений преимуществом является наличие синхронизации с системным генератором, что позволяет контроллеру памяти точно знать время готовности данных, благодаря чему временные задержки в процессе циклов ожидания уменьшаются, так как данные могут быть доступны во время каждого такта таймера.
- *ESDRAM (Enhanced SDRAM – улучшенная SDRAM)* является более быстрым вариантом архитектуры *SDRAM*. С точки зрения времени доступа производительность *ESDRAM* примерно в два раза выше.
- *FCRAM (Fast Cycle Random Access Memory – ОЗУ с быстрым циклом)*. Время выполнения цикла составляет всего 20 нс, то есть в 3 – 4 раза меньше, чем у современных модулей *DRAM*.
- *MRAM (Magnetic RAM – магнитное ОЗУ)* разработана компанией *Toshiba*, уже выпустившей опытный образец основной структуры кристалла, продемонстрировавшего выдающиеся скоростные качества — цикл чтения занимает всего 6 нс. Технология хранения информации в чипе *MRAM* заключается в размещении элемента, содержащего молекулы платины и кобальта, между двумя магнетопроводящими слоями. Запись и чтение происходят путем изменения магнитной активности в управляющих слоях.
- *DDR SDRAM (Double Data Rate SDRAM)* – синхронная динамическая память со случайным доступом и удвоенной скоростью передачи данных (возможности выборки 2 бит данных за один такт). Основное преимущество *DDR SDRAM* перед *SDRAM* то, что за один такт системного генератора могут выполняться две

операции с данными, что приводит к увеличению вдвое пиковой пропускной способности при работе на той же частоте. Применялась в *Pentium IV*, *Celeron* и *AMD Athlon/ Sempron*.

- *DDR2 SDRAM* аналогична *DDR*. Преимущества – возможности выборки 4 бит данных за один такт, более низкое энергопотребление, тепловыделение, а также увеличенная рабочая частота. Применяется в системах с процессорными разъемами *LGA 775* (*Pentium 4*, *Pentium Dual-Core*, *Core 2 Duo/Quad*) и с *AM2* (*AMD Phenom/ Athlon/Sempron*).
- *DDR3 SDRAM* – потомок *DDR2 SDRAM*, использует ту же технологию «удвоения частоты». Преимущество перед *DDR2* – способность работать на более высокой частоте, меньшее энергопотребление. *DDR3*, как и *DDR2*, имеют 240 контактов, но используют другие «ключи». Используется с *Intel Core i3/ 5/ 7* (*LGA 1366/ 1156*) и *AMD Phenom II/ Athlon II* (*AM3*).
- *RIMM (RDRAM, Rambus DRAM)* – синхронная динамическая память, разработанная компанией *Rambus*. Основные отличия от *DDR* памяти – увеличение тактовой частоты за счет уменьшения разрядности шины и одновременная передача номера строки и столбца ячейки при обращении к памяти.
- *SAM* – это память с последовательной выборкой. В такой памяти происходит последовательный доступ к ячейкам, вплоть до того, пока в конце концов не найдутся искомые данные. Она медленнее *RAM*, но хорошо подходит для буферной памяти. Данные в ней сохраняются последовательно в любом случае.

Постоянное запоминающее устройство

ПЗУ содержит информацию, которая не должна изменяться в ходе выполнения процессором вычислительных операций, например стандартные программы и константы. Эта информация заносится в ПЗУ перед установкой микросхемы в ЭВМ. Основные операции, которые может выполнять ПЗУ, – чтение и хранение.

Функциональные возможности ОЗУ шире, чем ПЗУ, но ПЗУ сохраняет информацию при отключении питания (т.е. является энергонезависимой памятью) и может иметь более высокое быстродействие, так как ограниченность функциональных возможностей ПЗУ и его специализация на чтении и хранении позволяют сократить время выполнения реализуемых им операций считывания.

Микросхемы ПЗУ также построены по принципу матричной структуры накопителя. Функции элементов памяти в них выполняют перемычки в виде проводников, полупроводниковых диодов или транзисторов. В такой матрице наличие перемычки может означать «1», а ее отсутствие – «0». Занесение информации в микросхему ПЗУ называется ее *программированием*, а устройство, с помощью которого заносится информация, – *программатором*. Программирование ПЗУ заключается в устранении (прожигании) перемычек по тем адресам, где должен храниться «0». Обычно схемы ПЗУ допускают только одно программирование, но специальные микросхемы – репрограммируемые ПЗУ (РПЗУ) – допускают их многократное стирание и занесение новой информации. Этот вид микросхем также относится к энергонезависимым, т.е. может длительное время сохранять информацию при выключенном питании.

Типы ПЗУ

Имеется четыре основных типа ПЗУ:

- ПЗУМ, *Read Only Memory (ROM)* – однократно программируемые изготовителем ПЗУ, называемые масочными;
- ППЗУ, *Programmable ROM (PROM)* – однократно программируемое пользователем постоянное запоминающее устройство;
- *EPROM (Erasable PROM)* – перепрограммируемые постоянные запоминающие устройства с ультрафиолетовым стиранием информации;
- *EEPROM (Electrically Erasable PROM)* – перепрограммируемые постоянные запоминающие устройства с электронным стиранием информации, также называемые *Flash ROM*.

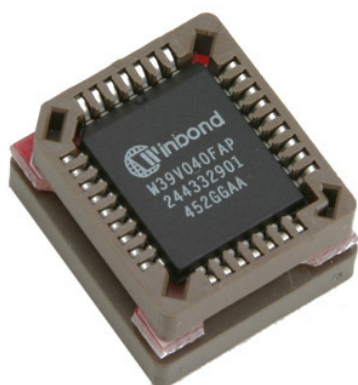


Рис. 7.7. Микросхема ПЗУ

Постоянное запоминающее устройство, которое еще называют встроенной программой, представляет собой интегральную микросхему (рис. 7.7), при изготовлении или эксплуатации запрограммированную определенными данными. ПЗУ используются не только в компьютерах, но и в большинстве других электронных устройств.

Размещение информации в основной памяти ЭВМ

Адресуемая единица информации основной памяти *IBM PC* – байт. Это означает, что каждый байт, записанный в ОП, имеет уникальный номер (адрес). При использовании 20-битной шины адреса абсолютный (физический) адрес каждого байта является пятиразрядным шестнадцатеричным числом, принимающим значения от 00000 до *FFFFFF* (рис. 7.8).

В младших адресах располагаются блоки операционной системы (векторы прерываний, зарезервированная область памяти *BIOS*), в этой же части могут размещаться драйверы устройств, дополнительные обработчики прерываний *DOS* и *BIOS*, командный процессор операционной системы. Затем располагается область памяти, отведенная пользователю. Область памяти пользователя заканчивается адресом *9FFFF*. Этот адрес является физической границей оперативного ЗУ, последним адресом 640-Кбайтной основной памяти.

Остальное адресное пространство (128 Кбайт с адреса *A0000* по *BFFFF*) отведено под видеопамять, которая физически размещается не в ОП, а в адаптере дисплея. После видеопамати расположено адресное пространство (256 Кбайт) постоянного запоминающего устройства, хранящего программы базовой системы ввода-вывода (*BIOS* – «*Basic Input/Output System*»).

Эта часть ОП еще называется *ROM-BIOS*. Из отведенных 256 Кбайт непосредственно ПЗУ занимает 64 Кбайта, а остальные 192 Кбайта оставлены для расширения ПЗУ. Поскольку большая часть оставленной для расширения *BIOS* части адресного пространства не используется, в этих адресах часто располагается информация, необходимая для работы сетевых карт, графических расширителей и др.

<i>FFFFFF</i> Расширения ПЗУ	256 Кб
ПЗУ (<i>ROM-BIOS</i>) <i>BFFFF</i>	
<i>BFFFF</i> Видеопамать <i>A0000</i>	128 Кб
<i>9FFFF</i> Область памяти, отведенной пользователю	640 Кб
Командный процессор операционной системы	
Драйверы устройств Векторы прерываний	
00000	

Рис. 7.8. Размещение информации в основной памяти

Расширение основной памяти ЭВМ

В начале 80-х годов Билл Гейтс, президент корпорации Microsoft, сказал: "Мне кажется, любому пользователю всегда будет достаточно объема оперативной памяти в 640 Кбайт..."

Рабочая концепция фирмы *IBM* при создании *IBM PC* также содержала гипотезу, что объем основной памяти ЭВМ, предназначенной для персонального использования в любой предметной области, не должен превышать 640 Кбайт. Поэтому в базовую модель *IBM PC* заложили 20-разрядную шину адреса системной магистрали. Наличие 20 линий в шине адреса позволяло адресовать память большего объема, чем было предусмотрено концепцией ($2^{20} = 1$ Мбайт). «Излишек» адресного пространства в 384 Кбайта был поделен между видеопамятью (128 Кбайт) и ПЗУ (256 Кбайт).

Физически увеличить объем памяти несложно, для этого необходимо только подключить к системной магистрали дополнительные модули. Такую возможность в *IBM PC* предусмотрели, но каждый байт дополнительной памяти должен иметь уникальный адрес, а адресного пространства для дополнительной памяти нет.

Существует несколько способов разрешения таких конфликтов. Один из них – *банкирование* памяти: вся память делится на блоки (банки), емкость которых не выходит за пределы допустимого адресного пространства; во время работы специальными командами можно переключать банки, делая активным любой из них, или осуществлять групповую перезапись информации из одного банка в другой.

В *IBM PC XT* фирма *IBM* применила другой способ: 256 Кбайт было сначала оставлено для ПЗУ, в котором размещалась базовая система ввода/вывода (*BIOS*). Анализ программ *BIOS* показал, что в оставленном для ПЗУ адресном пространстве (*UMB – Upper Memory Block*) имеются «окна» – неиспользуемые участки. Четыре таких участка (*page frames*) по 16 Кбайт были выделены, и их адреса стали использоваться для адресации дополнительной памяти, подключенной к системной магистрали. Таким образом общий объем ОП удалось уве-

личить на 64 Кбайта. Специальная программа (драйвер дополнительной памяти) «перехватывала» обращение к «окнам» ПЗУ и вместо них «подставляла» дополнительный модуль памяти (*Expanded Memory*).

Дополнительная память не обязательно должна была иметь объем 64 Кбайта. Ее объем мог быть и значительно большим (выпускаются модули дополнительной памяти объемом 32, 64 Мбайт и более). При этом драйвер дополнительной памяти делил ее на блоки по 16 Кбайт и «отображал» каждое окно *UMB* на один из блоков *Expanded Memory*. Из-за этого память такого вида получила название *отображаемой памяти*.

Но развитие персональных ЭВМ привело к необходимости более серьезной корректировки рабочей концепции. Увеличение разрядности шины адреса и адресных регистров до 32 и 64 изменило принцип формирования абсолютного адреса ОП.

Виртуальные режимы отображения памяти ЭВМ

Желание использовать в реальном режиме всю фактически имеющуюся в наличии дополнительную память привело к созданию двух виртуальных режимов:

1. *Стандарт EMS (Expanded Memory Specifications)*, реализующий принцип банкирования дополнительной памяти. Вся дополнительная память делится на страницы (банки) емкостью по 16 Кбайт; выбираются четыре страницы и объявляются активными. Выбранные активные страницы отображаются на четыре окна *UMB*, теперь при обращении к одному из окон *UMB* вместо него подставляется отображенная на него страница дополнительной памяти. Это позволило использовать всю дополнительную память любого объема.

В соответствии с этим стандартом работают драйверы *XMA2EMS.SYS*, *EMM386.SYS* и др.

2. *Стандарт XMS (Extended Memory Specification)*. Это другой виртуальный режим, основанный на том, что за счет кратковременного разблокирования дополнительных (по сравнению с *XT*) линий шины адреса системной магистрали удается увеличить доступное *MS DOS* адресное пространство еще почти на 64 Кбайта, начиная с адреса

FFFFFF (т.е. за пределами адресного пространства в 1 Мбайт). Эта область адресного пространства (64 Кбайта, начиная с 1 Мбайта) получила название *HMA (High Memory Area)* – старшая область памяти.

Блоки памяти, расположенные выше границы *HMA*, называются *EMB (Extended Memory Blocks)* – расширенные блоки памяти.

В этом стандарте программные модули могут располагаться только в *HMA*, а остальная память может использоваться лишь для хранения данных. Стандарт *XMS* реализуется драйвером *HIMEM.SYS*, который способен работать с шиной адреса, имеющей до 32 линий.

Сверхоперативные запоминающие устройства

Используются для хранения небольших объемов информации и имеют значительно меньшее время (в 2 – 10 раз) считывания/записи, чем основная память. *СОЗУ* обычно строятся на регистрах и регистровых структурах.

Регистр представляет собой электронное устройство, способное хранить занесенное в него число неограниченно долго (при включенном питании). Наибольшее распространение получили регистры на статических триггерах.

По назначению регистры делятся на регистры хранения и сдвига. Информация в регистры может заноситься и считываться либо параллельно, сразу всеми разрядами, либо последовательно, через один из крайних разрядов с последующим сдвигом занесенной информации.

Сдвиг записанной в регистр информации может производиться вправо или влево. Если регистр допускает сдвиг информации в любом направлении, он называется *реверсивным*.

Регистры могут быть объединены в единую структуру. Возможности такой структуры определяются способом доступа и адресации регистров.

Если к любому регистру можно обратиться для записи/чтения по его адресу, такая регистровая структура образует *СОЗУ с произвольным доступом*.

Безадресные регистровые структуры могут образовывать два вида устройств памяти: магазинного типа и память с выборкой по содержанию (ассоциативные *ЗУ*).

Память *магазинного типа* образуется из последовательно соединенных регистров (рис. 7.9).

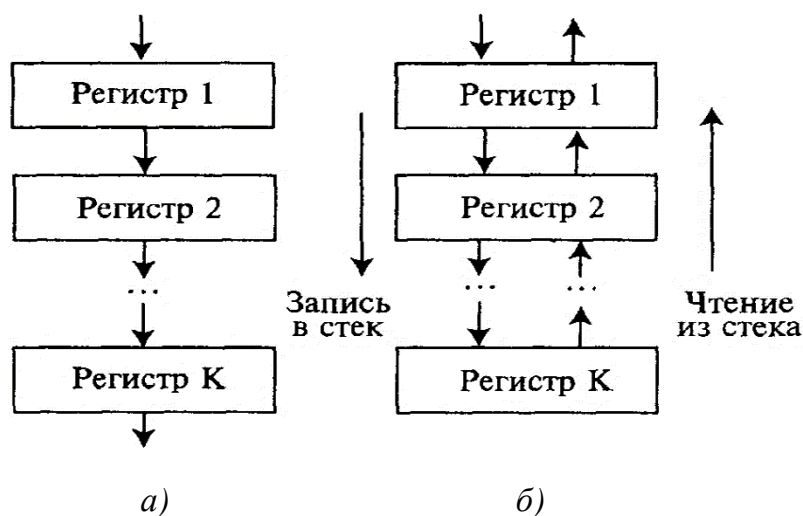


Рис. 7.9. Регистровая структура магазинного типа:
 а – типа *FIFO*; б – типа *FILO*

Если запись в регистровую структуру (рис. 7.9, а) производится через один регистр, а считывание – через другой, то такая память является аналогом линии задержки и работает по принципу «первым вошел – первым вышел» (*FIFO – first input – first output*).

Если же запись и чтение осуществляются через один и тот же регистр (рис. 7.9, б), такое устройство называется *стековой памятью*, работающей по принципу «первым вошел – последним вышел» (*FILO – first input – last output*). При записи числа в стековую память сначала содержимое стека сдвигается в сторону последнего, К-го регистра (если стек был полностью заполнен, то число из К-го регистра теряется), а затем число заносится в вершину стека – регистр 1. Чтение осуществляется тоже через вершину стека. После того как число из вершины прочитано, стек сдвигается в сторону регистра 1.

Стековая память получила широкое распространение. Для ее реализации в ЭВМ разработаны специальные микросхемы. Но часто работа стековой памяти *эмулируется* в основной памяти ЭВМ. В стек может быть загружен в определенной последовательности ряд данных, которые впоследствии считываются из стека уже в обратном порядке, на этом свойстве построена система арифметических преобразований информации, известная под названием «логика Лукашевича».

Память *с выборкой по содержанию* является безадресной. Обращение к ней выполняется по специальной маске, которая содержит поисковый образ. Информация считывается из памяти, если часть ее соответствует поисковому образу, зафиксированному в маске. Например, если в такую память записана информация, содержащая данные о месте жительства (включая город), и необходимо найти сведения о жителях определенного города, то название этого города помещается в маску и дается команда *чтение* – из памяти выбираются все записи, относящиеся к заданному городу.

В микропроцессорах ассоциативные ЗУ используются в составе *кэш-памяти* для хранения адресной части команд и операндов, исполняемой программы. При этом нет необходимости обращаться к ОП за следующей командой или требуемым операндом: достаточно поместить в маску необходимый адрес; если искомая информация имеется в СОЗУ, она будет сразу выдана. Обращение к ОП будет необходимо лишь при отсутствии требуемой информации в СОЗУ. За счет такого использования СОЗУ сокращается число обращений к ОП, а это позволяет экономить время, так как обращение к СОЗУ требует в 2 – 10 раз меньше времени, чем обращение к ОП.

Кэш-память может быть размещена в кристалле процессора (так называемая «кэш-память I уровня») или выполнена в виде отдельной микросхемы (внешняя кэш-память, или кэш-память II уровня). Встроенная кэш-память (I уровня) в процессорах *Pentium* имеет объем около 16 Кбайт, время доступа 5 – 10 нс, работает с 32-битными словами и обеспечивает пропускную способность более 667 Мбайт/с. Внешняя кэш-память (II уровня) имеет объем 256 Кбайт – 1 Мбайт, время доступа – 15 нс, работает с 64-битными словами и обеспечивает максимальную пропускную способность более 528 Мбайт/с. Конструктивно выполняется либо в виде 28-контактной микросхемы, либо в виде модуля расширения на 256 или 512 Кбайт.

В современных ЭВМ микросхемы памяти (ОП и СОЗУ) изготавливают из кремния по полупроводниковой технологии с высокой степенью интеграции элементов на кристалле (микросхемы памяти относятся к так называемым «регулярным» схемам, что позволяет сделать установку элементов памяти в кристалле (чипе) настолько плотной, что размеры элементов памяти становятся сопоставимыми с размерами отдельных атомов).

7.2.2. Внешние запоминающие устройства

Внешняя память предназначена для длительного хранения программ и данных, и целостность ее содержимого не зависит от того, включен или выключен компьютер. В отличие от оперативной памяти *внешняя память не имеет прямой связи с процессором*. Информация от внешнего запоминающего устройства (ВЗУ) к процессору и наоборот циркулирует примерно по следующей цепочке:

ВЗУ \longleftrightarrow ОЗУ \longleftrightarrow Кэш \longleftrightarrow Процессор

В состав внешней памяти компьютера входят:

- магнитные накопители;
- оптические накопители;
- магнито-оптические накопители.

Магнитные накопители

1. *Накопители на гибких магнитных дисках* используют *гибкий диск* (англ. *Floppy disk*), или дискету – носитель небольшого объема информации, представляющий собой гибкий пластиковый диск в защитной оболочке.

На диски нанесены концентрические дорожки. Запись и хранение информации на этих пластинах происходит за счёт преобразования электрических сигналов в определённые изменения магнитного поля с последующим воздействием этим полем на магнитную пластину. Благодаря явлению остаточного магнетизма следы от этих воздействий сохраняются в магнитном материале на длительный срок. Считывание информации, т.е. воспроизведение электрических сигналов, происходит точно так же, только в обратном направлении.

Магнитные домены или битовые ячейки представляют собой чередующиеся участки с различным направлением намагниченности. Плотность магнитной пластины определяется размерами ячеек: чем они меньше, тем выше плотность записи информации.

Битовые ячейки формируют секторы, которые впоследствии определяют минимальную логическую единицу хранения данных – кластер (рис. 7.10).

Для считывания и записи информации используются магнитные головки, которые собраны на механическом перемещающемся приводе, предназначенном для позиционирования.

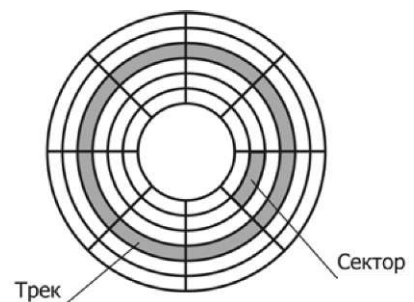


Рис. 7.10. Поверхность магнитного диска

Информация записывается по концентрическим дорожкам (*трекам*), которые делятся на *секторы*. Количество дорожек и секторов зависит от типа и формата дискеты. Сектор хранит минимальную порцию информации, которая может быть записана на диск или считана. Емкость сектора постоянная и составляет 512 байт.

Характеристики дискет: диаметр 3,5 дюйма (89 мм), емкость 1,44 Мбайта, число дорожек 80, количество секторов на дорожках 18.

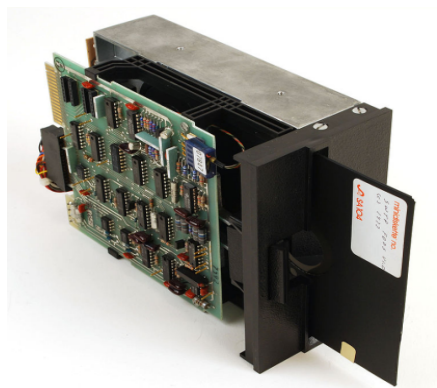


Рис. 7.11. Конструкция НГМД

Дискета устанавливается в накопитель на гибких магнитных дисках (НГМД) (англ. *Floppy-disk drive*), конструкция которого показана на рис. 7.11. Дискета автоматически в нем фиксируется, после чего механизм накопителя раскручивается до частоты вращения 360 об/мин. В накопителе вращается сама дискета, магнитные головки остаются неподвижными.

Дискета вращается только при обращении к ней. Накопитель связан с процессором через контроллер гибких дисков.

2. *Накопитель на жестких магнитных дисках* (англ. *HDD – Hard Disk Drive*), или *винчестер*, – это наиболее массовое запоминающее устройство большой емкости, в котором носителями информации являются круглые пластины, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения информации – программ и данных.



Рис. 7.12. Конструкция НЖМД

Накопитель на жестком магнитном диске (НЖМД) имеет тот же принцип действия, что и НГМД, но отличается тем, что магнитный носитель информации в нем несъемный и состоит из нескольких пластин, закрепленных на общей оси пакета магнитных носителей.

Количество дисков, каждый из которых имеет по две рабочие поверхности, в накопителе может быть от 3 до 10 и более (рис. 7.12).

Сами диски представляют собой обработанные с высокой точностью керамические или алюминиевые пластины, на которые с помощью технологии напыления нанесен специальный магнитный слой.

Накопители на жестких магнитных дисках делают герметичными – малое расстояние (зазор) между рабочей поверхностью и магнитной головкой должно быть защищено от пылинок, чтобы уберечь тонкий напыленный слой кобальта от стирания. Магнитная головка во время работы не должна касаться поверхности диска и в то же время находиться от нее на расстоянии в десятые доли микрона.

Каждую рабочую поверхность диска обслуживает своя головка, которая во время записи/чтения информации неподвижна. Магнитный «след» на поверхности диска, образовавшийся при работе головки на запись, образует кольцевую траекторию – *дорожку (track)*. Дорожки, расположенные друг под другом на всех рабочих поверхностях магнитного носителя, называются *цилиндром*.

Магнитные головки при работе НЖМД могут перемещаться, настраиваясь на требуемую дорожку. Перед началом эксплуатации пакет магнитных дисков форматируется: на нем размечаются дорожки (ставится маркер начала дорожки и записывается ее номер), наносятся служебные зоны секторов на дорожках. Для записи/чтения информации контроллеру НЖМД передается адрес: номер цилиндра, номер рабочей поверхности цилиндра, номер сектора на выбранной дорожке. На основании этого магнитные головки перемещаются к нужному цилиндру, ожидают появления маркера в начале дорожки, появления требуемого сектора, после чего записывают или читают информацию из него. Несмотря на то что все магнитные головки установлены на требуемый цилиндр, работает в каждый данный момент только одна головка.

Внутри любого винчестера обязательно находится печатная плата с электронными компонентами. Эта электроника расшифровывает команды контроллера жесткого диска, стабилизирует скорость вращения двигателя, генерирует сигналы для головок записи и усиливает их от головок чтения и т.п.

У современных моделей скорость вращения шпинделя (вращающего вала) обычно составляет 7200 об/мин, среднее время поиска дан-

ных – 9 мс, средняя скорость передачи данных – до 60 Мбайт/с. В отличие от дискеты жесткий диск *вращается непрерывно*. Все современные накопители снабжаются *встроенным кэшем* (обычно 2 Мбайта), который существенно повышает их производительность. Винчестер связан с процессором через *контроллер жесткого диска*.

Ёмкость жёсткого диска напрямую связана с плотностью и количеством пластин. Чем больше плотность и количество пластин, тем больше объём жёсткого диска. Однако повышать ёмкость исключительно за счёт увеличения количества пластин бессмысленно. Во-первых, корпус обыкновенного 3,5-дюймового винчестера способен уместить максимум 5 пластин и 10 головок. Во-вторых, большое количество пластин и головок увеличивает энергопотребление и тепловыделение, что повышает риск аппаратного сбоя из-за большого числа подвижных элементов.

Для увеличения линейной плотности записи информации необходимо максимально уменьшать длину битовых ячеек и делать переходы между ними максимально резкими. Однако с уменьшением длины у битовой ячейки снижается устойчивость к внешним магнитным полям, в результате чего возникает так называемый супермагнетизм. Длина битовой ячейки уменьшается до критической отметки, и размагничивающиеся поля становятся настолько большими, что ячейка саморазмагничивается и исчезает. Иначе говоря, происходит самопроизвольное стирание данных.

Основные игроки рынка винчестеров смогли решить эту проблему. Благодаря технологии перпендикулярной магнитной записи *PMR (Perpendicular Magnetic Recording)* производителям жёстких дисков удалось получить плотность в 200 Гбайт для одной пластины. Перпендикулярное расположение магнитных доменов позволило достигнуть высокой плотности без проявления суперпарамагнитного эффекта.

Большинство жёстких дисков выпускается для двух интерфейсов – *SATA* и *PATA*. Их пропускная способность составляет 300 Мбит/с (*Serial ATA II*) и 133 Мбит/с соответственно. Объём кэш-памяти большинства современных жёстких дисков составляет 8 и 16 Мбайт, хотя встречаются на рынке модели с большим объёмом кэша.

Жёсткие диски выпускаются также в виде внешних *USB-HDD* накопителей 3,5 дюйма (*Portable Storage Device*). Пример – жесткий внешний диск 1,5 Tb *WD My Book World Edition II (WDG2NC15000E)*, 7200 rpm, 16 mb, Ethernet, NOT USB.

Дистанционный доступ к данным, хранящимся на компьютере, обеспечивается посредством подключения накопителя *HDD* к кабельной или беспроводной сети.

Внешние *HDD USB* позволяют объединять диски в зеркальные *RAID*-массивы (рис. 7.13) для надежной защиты данных, они имеют три интерфейса: (*FireWire 400/800, USB 2.0*), делает винчестер высокоскоростным.



Рис. 7.13. RAID-массив

Накопители на оптических дисках (оптический привод)

Оптический привод представляет собой устройство хранения данных с оптическим принципом считывания и записи. В качестве носителей оптический привод использует плоские многослойные диски диаметром 8 или 12 мм.

Типы оптических приводов ***CD-ROM***

Это компакт-диск (*Compact Disk Read-Only Memory*), с которого можно только читать. Самое простое из устройств подобного типа.

CD-ROM представляет собой прозрачный полимерный диск диаметром 12 см и толщиной 1,2 мм, на одну сторону которого напылен светоотражающий слой алюминия, защищенный от повреждений слоем прозрачного лака.

Информация на диске представляется в виде последовательности *впадин* (углублений в диске) и *выступов* (их уровень соответствует поверхности диска), расположенных на спиральной дорожке, выходящей из области вблизи оси диска (рис. 7.14).

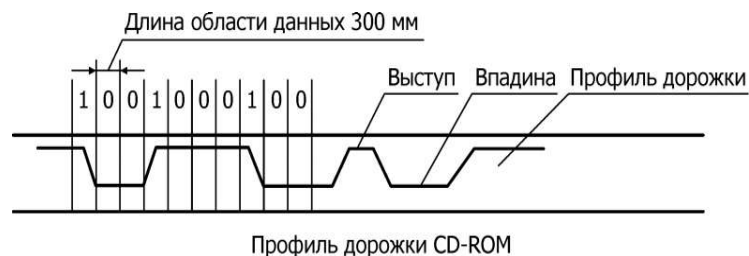


Рис. 7.14. Представление информации на оптическом диске

В отличие от магнитных дисков компакт-диски имеют не множество кольцевых дорожек, а одну – *спиральную*, как у грампластинок.

Накопитель CD-ROM преобразует последовательность углублений и выступов на поверхности в последовательность двоичных сигналов. Для этого используется *считывающая головка с микролазером и светодиодом*.



Рис. 7.15. Привод оптического диска

Скорость большинства «современных» *CD-ROM* достигает 52x, реже максимальных для данного типа устройств 56x. На сегодняшний день привод *CD-ROM* (рис. 7.15) морально устарел и представляет интерес в самых исключительных случаях. Привлекательности в *CD-ROM* нет никакой,

компакт-диски *CD* стоят ненамного дешевле обычных *DVD*, а их ёмкость значительно меньше.

CD-RW

Это следующий этап развития оптических приводов. *CD-RW* позволяет не только считывать информацию с обычных компакт-дисков, но и записывать её на матрицы *CD*.

Накопители *CD-RW* позволяют многократно записывать информацию на диски с отражающей поверхностью, под которую нанесен слой пластика типа *Ag-In-Sb-Te* (содержащего *silver, indium, antimony, tellurium*) с изменяемой фазой состояния. Фаза пластика кристаллическая или аморфная, изменяется в зависимости от скорости остывания после разогрева поверхности лазерным лучом в процессе записи, выполняемой непосредственно в дисководе ПК.

При медленном остывании пластик переходит в кристаллическое состояние, и информация стирается (записывается «0»); при быстром остывании (если разогрета только микроскопическая точка), элемент пластика переходит в аморфное состояние (записывается «1»). Ввиду разницы коэффициентов отражения от кристаллических и аморфных микроскопических точек активного слоя при считывании происходит модуляция интенсивности отраженного луча, воспринимаемого головкой чтения.

Актуальность *CD-RW* в настоящее время также под большим вопросом, только офисное применение, и то в исключительных случаях.

DVD-ROM

Это цифровой видеодиск (*Digital Video Disk*). Ещё один этап эволюции оптических приводов – устройство, способное читать не только обычные *CD*-диски, но и компакт-диски *DVD*.

В *DVD* используется однослойная и двухслойная, односторонняя и двухсторонняя уплотненная запись. Уплотнение записи данных на *DVD* было достигнуто путем уменьшения диаметра пишущего/читающего луча (зелено-голубой лазер) в два раза, при этом уменьшаются сами точки (питы), уменьшается расстояние между соседними точками на дорожке и увеличивается количество дорожек. За счет увеличения плотности записи, более эффективной битовой модуляции каналов, более эффективного исправления ошибок, меньшего перекрытия секторов емкость по сравнению с *CD* увеличилась в 7 раз. Стандартный однослойный односторонний диск *DVD* может хранить 4,7 Гбайта данных, двухслойный накопитель имеет емкость 8,5 Гб. Скоростная формула устройства выглядит следующим образом: 16x для *DVD* и 52x для *CD*.

DVD-CD-RW Combo

Так называемый *Combo*-драйв, который сочетает в себе функции таких устройств, как *DVD-ROM* и *CD-RW* и соответственно может записывать диски *CD-R* и *CD-RW*, считывать как обычные *CD*, так и *DVD*.

Blu-Ray

Здесь за основу технологической разработки был принят новый стандарт, предусматривающий считывание информации с помощью синего лазера. Для обычного *CD-ROM* применялся луч лазера с длиной волны 780 нм, для технологии *DVD* стали применять лазер с более короткой длиной волны 650 или 635 нм, и это увеличило емкость. С учетом двухслойности емкость *DVD*-дисков достигла 15,9 Гб.

В технологии *Blu-Ray* используется синий лазер с длиной волны 405 нм. Такое уменьшение позволило сузить дорожку в 2 раза в сравнении с обычным *DVD*. Уменьшение толщины защитного слоя в 6 раз (до 0,1 мм) увеличило качество чтения/записи. Увеличение числовой

апертуры линзы с 0,6 до 0,85 позволило повысить плотность записи в 2 раза, а более короткая длина волны – в 2,6 раза, что позволило увеличить емкость по сравнению с *DVD* в $2 \times 2,6 = 5,2$ раза. В настоящее время имеются диски с удвоенным значением емкости 54 Гб и более.

Накопители на магнитооптических дисках

Принцип работы магнитооптического накопителя (*Magneto Optical*) основан на использовании двух технологий – лазерной и магнитной. Запись информации осуществляется на магнитном носителе, а оптический лазерный луч используется для местного разогрева точки магнитной поверхности. Сущность процессов чтения-записи обусловлена следующим: активный слой на поверхности магнитооптического диска может быть перемагничен магнитной головкой только при высокой температуре. Такая температура (сотни градусов) создается лазерным импульсом длительностью порядка 0,1 мкс. При считывании информации вектор поляризации отраженного от поверхности диска лазерного луча на несколько градусов изменяет свое направление в зависимости от направления намагниченности элемента активного слоя. Изменение направления поляризации и воспринимается соответствующим датчиком.

Существуют два типа магнитооптических накопителей:

- с однократной записью стандарта *CC-WORM (Write Once Read Many)*;
- перезаписываемые *CC-E (Continuous Composite Erasable)* стандарта *LIMDOW (Light Intensity Modulation/Direct OverWrite)*.

К основным недостаткам стандартной магнитооптической технологии относится прежде всего низкая скорость перезаписи, поскольку данный процесс требует выполнение трех циклов – стирания старых данных, записи новых и проверки. Для уменьшения времени перезаписи цикл проверки, как правило, не используется. Чтобы еще более увеличить скорость перезаписи, была разработана спецификация *Overwrite*, которая устраняет цикл стирания. Диски стандарта *LIMDOW* совместимы с этой спецификацией и, таким образом, позволяют повысить суммарное быстродействие.

В магнитооптических накопителях *CC-WORM* для предотвращения стирания и повторной записи информации на диск на контрольные дорожки наносятся специальные метки.

Магнитооптические накопители имеют два типоразмера: 3,5" и 5,25". Магнитооптические диски форм-фактора 3,5" могут иметь следующие емкости: 128 Мбайт, 230 Мбайт, 540 Мбайт, 640 Мбайт и 1,3 Гбайт; эти диски являются односторонними.

Время доступа у магнитооптических накопителей находится в пределах от 50 до 150 мс, скорость считывания до 3000 Кбайт/с. Магнитооптические накопители в ПК могут быть внутренними и внешними, последние предпочтительнее ввиду значительного тепловыделения. Магнитооптические диски позволяют переносить большие объемы данных и отличаются высокой степенью надежности. Однако в силу относительно высокой стоимости дисководов и дисков их область применения ограничена профессиональными системами обработки графики, видеомонтажа, верстки и тому подобное, когда требуются накопление и обмен большими объемами данных. Магнитооптические накопители могут также использоваться для решения задач резервного копирования. Большинство магнитооптических накопителей имеет интерфейс *SCSI*.

USB-накопители на основе флэш-памяти

Основные достоинства – простота в эксплуатации, достаточно широкий модельный ряд и небольшая стоимость. На сегодняшний день данный носитель если не заменил, то успешно конкурирует с другими типами носителей информации. Во флэш-драйве используется особый вид памяти – *flash* (флэш-память). Этот тип памяти энергонезависим – после отключения все записанное в нее сохраняется.

Основные различия между флэш-накопителями заключаются в дизайне (флэш-накопитель со съемным футляром (рис. 7.16, а), с вращающимся футляром (рис. 7.16, б), с утапливаемой контактной площадкой (рис. 7.16, в)), емкости, типах интерфейса и дополнительных возможностях.



Рис. 7.16. Примеры корпусов флэш-*USB*-накопителей

С помощью программного обеспечения можно защитить паролем хранящиеся на флэш-драйве данные. На некоторых моделях флэш-драйвов имеется специальный переключатель, который разрешает или запрещает запись и удаление информации.

Несложно выделить флэш-*USB*-накопители, которые демонстрируют высокую производительность при чтении и записи файлов: *KingSton DataTraveler II Plus Migo Edit* и *OCZ Rally2* – несомненные лидеры. *Apacer Handy Steno AH421* и *KingSton DataTraveler II* демонстрируют хорошие средние результаты, а *Transcend JetFlash 185*, *Transcend JetFlash V20* и *Imation Pocket Flash Drive* обладают, что называется, «бюджетными» скоростными показателями, т.е. уровень производительности при чтении и записи файлов у данных *USB*-накопителей на основе флэш-памяти соответствует их цене.

Вопросы к компьютерному тестированию

1. Как называют устройство ЭВМ, предназначенное для преобразования представления данных и управляющих сигналов в разных устройствах ПК?
2. Как называют устройство ЭВМ, предназначенное для управления работой механических подвижных частей устройства и формирования электрических импульсов при записи и чтении?
3. Как называют порт, предназначенный для побайтной передачи данных за один такт машинного времени?
4. Какова емкость памяти матрицы ОЗУ при числе линий системной магистрали $m = 8$?
5. Что используют в памяти матрицы *DRAM* в качестве запоминающего элемента?
6. Какие форм-факторы модулей памяти могут использоваться в компьютерах с процессорами *Pentium* только парами?
7. Какие форм-факторы модулей памяти используются в ноутбуках и субноутбуках?
8. Какой из типов памяти использует принцип записи и чтения путем изменения магнитной активности в управляющих слоях?
9. Какой из типов памяти имеет возможность выборки 2 бит данных за один такт?
10. Какой из типов памяти имеет последовательный доступ к ячейкам и великолепно подходит для буферной памяти?
11. Какой из типов ПЗУ является однократно программируемым пользователем?

12. Какой из типов ПЗУ основан на принципах, используемых в *flash*-памяти?
13. Если обращение к регистру памяти осуществляется по специальной маске, которая содержит поисковый образ, то что образует такая регистровая структура?
14. Как называют память, в которой запись и чтение осуществляются через один и тот же регистр?
15. Что содержит маска стековой памяти?
16. Что является адресуемой единицей информации основной памяти *IBM PC*?
17. Где физически размещается видеопамять?
18. Как называется способ разрешения конфликта, когда вся память делится на блоки, емкость которых не выходит за пределы допустимого адресного пространства?
19. Какой из стандартов реализует принцип кратковременного разблокирования дополнительных линий шины адреса системной магистрали с целью увеличения доступного для *MS DOS* адресного пространства?
20. Какова форма дорожек на диске при магнитной записи?
21. Какова форма дорожек на диске при оптической записи с постоянной линейной скоростью?
22. В приводах каких накопителей двигатель включается только при доступе по чтению или записи?
23. Какой способ записи используют в накопителях сверхвысокой плотности?
24. Соприкасается ли магнитная головка НЖМД с поверхностью диска?
25. Как называют устройство, объединяющее несколько запоминающих устройств в один большой накопитель?
26. Какую технологию записи и фазу состояния чувствительного материала используют при записи сигнала «1» на компакт-диск *CD-RW*?
27. За счет чего достигается высокая плотность записи данных в *DVD*-дисководах?
28. В каком виде представлены «1» и «0» на поверхности диска *CD-RW*?
29. Какую длину волны используют лазеры в приводах дисков по технологии *Blu-Ray*?
30. С какой целью используют лазерный луч при магнитооптической записи?



Глава 8

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

Рассматриваемые вопросы:

- 8.1. Структура программного обеспечения ЭВМ.
- 8.2. Общее программное обеспечение.
- 8.3. Операционные системы.
- 8.4. Системы автоматизации программирования.
- 8.5. Комплекс программ технического обслуживания.
- 8.6. Специальное программное обеспечение. Пакеты программ.

8.1. Структура программного обеспечения ЭВМ

Программное обеспечение ЭВМ разделяют на *общее, или системное (general Software)*, и *специальное, или прикладное (application or special Software)* [15].

Общее ПО объединяет программные компоненты, обеспечивающие многоцелевое применение ЭВМ и мало зависящие от специфики вычислительных работ пользователей. Сюда входят программы, организующие вычислительный процесс в различных режимах работы машин, программы контроля работоспособности ЭВМ, диагностики и локализации неисправностей, программы контроля заданий пользователей, их проверки, отладки и т.д.

Общее ПО обычно поставляется потребителям комплектно с ЭВМ. Часть этого ПО может быть реализована в составе самого компьютера. Например, в ПЭВМ часть программ ОС и часть контролирующих тестов записана в ПЗУ этих машин.

Специальное ПО (СПО) содержит пакеты прикладных программ пользователей (ППП), обеспечивающие специфическое применение ЭВМ и вычислительной системы (ВС).

Прикладной программой называется программный продукт, предназначенный для решения конкретной задачи пользователя. Обычно прикладные программы объединяются в пакеты (например, пакеты для управления сложными технологическими процессами, разработки экономических документов, рекламных роликов, планирования и др.)

8.2. Общее программное обеспечение

Общее ПО включает в свой состав операционную систему (ОС), средства автоматизации программирования (САП), комплекс программ технического обслуживания (КПТО), пакеты программ, дополняющие возможности ОС (ППос), и систему документации (СД).

Операционная система служит для управления вычислительным процессом путем обеспечения его необходимыми ресурсами.

Средства автоматизации программирования объединяют программные модули, обеспечивающие этапы подготовки задач к решению.

Модули КПТО предназначены для проверки работоспособности вычислительного комплекса.

Важной частью ПО является *система документации*, хотя она и не является программным продуктом. СД предназначается для изучения программных средств, она определяет порядок их использования, устанавливает требования и правила разработки новых программных компонентов и особенности их включения в состав общего или специального программного обеспечения.

Программное обеспечение современных ЭВМ и ВС строится по иерархическому модульному принципу (рис. 8.1). Это обеспечивает возможность адаптации ЭВМ и ВС к конкретным условиям применения, открытость системы, способность систем к совершенствованию и т.д.

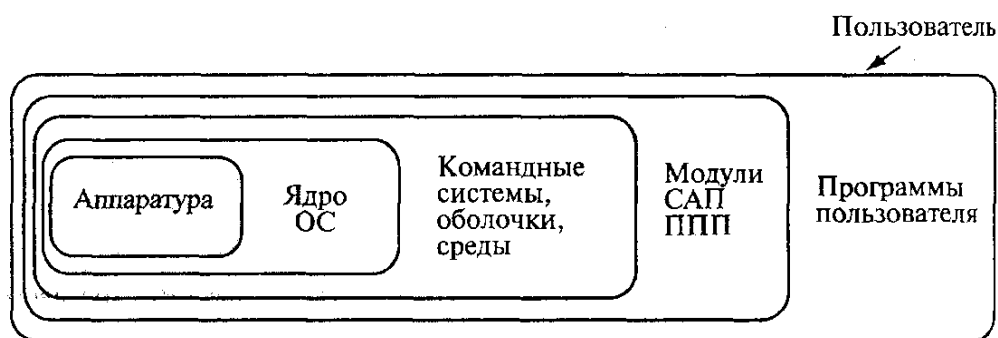


Рис. 8.1. Иерархия программных средств ЭВМ

Нижний уровень образуют *программы пользователя ОС*, которые играют роль посредника между техническими средствами системы и пользователем. Однако прямое использование команд ОС требует от пользователя определенных знаний и специальной компьютерной подготовки. Поэтому на практике пользователи, как правило, работают не напрямую с ОС, а через *командные системы* – пакеты программ, дополняющие возможности ОС (ППос).

Ярким примером подобных систем могут служить пакеты *Total (Norton, Volkov) Commander, DOS Navigator* и другие, завоевавшие заслуженную популярность у пользователей. С помощью этих систем трудоемкость работы с компьютером значительно сокращается. Работа пользователя при этом заключается в выборе определенных рубрик меню.

Развитие и усложнение средств обработки ОС и командных систем привело к появлению *операционных сред* (например, *Microsoft Windows*), обеспечивающих графический интерфейс с широчайшим спектром услуг.

Квалифицированные пользователи, разрабатывающие собственные программные продукты, используют компоненты САП.

Программы КПТО непосредственного участия в вычислениях не принимают, они только обеспечивают их. Перед началом вычислений их задача – проверка работоспособности аппаратуры.

8.3. Операционные системы

Центральное место в структуре ПО занимает операционная система. Она представляет собой "систему программ, предназначенную для обеспечения определенного уровня эффективности цифровой вычислительной системы за счет автоматизированного управления ее работой и предоставляемого пользователям набора услуг" (ГОСТ 15971-84).

Программные компоненты ОС обеспечивают управление вычислениями и реализуют такие функции, как планирование и распределение ресурсов, управление вводом-выводом информации, управление данными.

Цель применения ОС:

- увеличение быстродействия и пропускной способности ЭВМ;
- контроль работоспособности технических и программных средств;
- облегчение работы операторов при использовании ими технических и программных средств;
- управление программами и данными в ходе вычислений;
- обеспечение структурной гибкости ЭВМ, заключающейся в способности изменяться, пополняться новыми техническими и программными средствами.

Конкретная конфигурация ОС формируется на нескольких уровнях. Предварительно этот состав определяется при генерации ОС. "*Генерация системы* – это процесс выделения отдельных частей операционной системы и построения частных операционных систем, отвечающих требованиям системы обработки данных" (стандарт ISO 2382/10-79). Из полного набора программных модулей ОС (*дистрибутива*) формируется специальный набор этих средств, в наибольшей степени отвечающий запросам пользователей. Коррекция же состава используемых услуг может быть выполнена непосредственно перед решением задач операторами вычислительного центра или самими пользователями. Оперативное обращение к средствам ОС возможно и из программ пользователей путем включения в них специальных директив.

Применительно к IBM PC для каждого ПК создается так называемый *системный диск* с соответствующим набором программ дисковой операционной системы (ДОС). При включении компьютера программой *BIOS Setup Program* (программа начальных установок) обеспечивается запись основных параметров системы, которые сохраняются в *CMOS*-памяти компьютера. Окончательная настройка ДОС производится файлами *autoexec.bat* и *config.sys*, а также выполнением отдельных команд, набираемых в командной строке ДОС.

Для каждого типа ЭВМ возможно использование нескольких типов ОС. Все они имеют несколько версий. Для *IBM PC* распространение получили *MS DOS* фирмы *Microsoft*, *OS/2 Warp* и *DOS* фирмы ЮМ, *DR DOS* фирмы *Digital Research*, *DOS* фирмы *Novell*. Отличия ОС опре-

деляются составом и детализацией системных функций. Более распространенной является *MS DOS*, она используется в большинстве компьютеров. Система *DR DOS* имеет развитые средства защиты информации и разграничение доступа, что предопределяет ее использование в системах с закрытием обрабатываемой информации. *OS/2 Warp* позволяет более полно использовать возможности самых мощных микропроцессоров при организации вычислительного процесса, *DOS* фирмы *Novell* ориентирована на работу ЭВМ в сети.

Основу любой ОС составляет *управляющая программа*. Основные функции ее: управление заданиями, управление задачами – ходом выполнения отдельных программ и управление данными.

Задание – это требование пользователя по выполнению некоторого объема вычислительных работ. Процедуры управления заданиями обеспечивают предварительное планирование работы ЭВМ и оперативную связь пользователя и оператора с машиной во время работы. Планирование работы включает: ввод пакетов или одиночных заданий, формирование очередей заданий в соответствии с их приоритетами, активизацию (запуск) и завершение заданий.

Задачи образуют отдельные программы вместе с обрабатываемыми ими данными (например, трансляции, редактирования).

Управление задачами требует *распределения и назначения ресурсов*, а именно управления временем работы процессора, распределения оперативной памяти, управления очередностью задач и пр. Ведущая программа управления задачами – управляющая программа-диспетчер: супервизор, базовый модуль ДОС или др. Часто используемые модули образуют *ядро ОС*, которое постоянно находится в оперативной памяти и быстро реагирует на изменяющиеся условия функционирования. Примером такой программы может служить командный процессор *command.com*.

Набор программ *управления данными* обеспечивает процессы организации, идентификации, размещения в ОП и на ВЗУ, хранения, построения библиотек и выборки всех данных, которые могут обрабатываться в ЭВМ.

Структурно ОС *IBM PC* состоит из следующих элементов, представленных на рис. 8.2. Кроме программных компонентов, указанных на рисунке, к ДОС относят еще вспомогательные файлы *autoexec.bat* и *config.sys*. Они предназначаются для настройки на конкретные режимы работы.

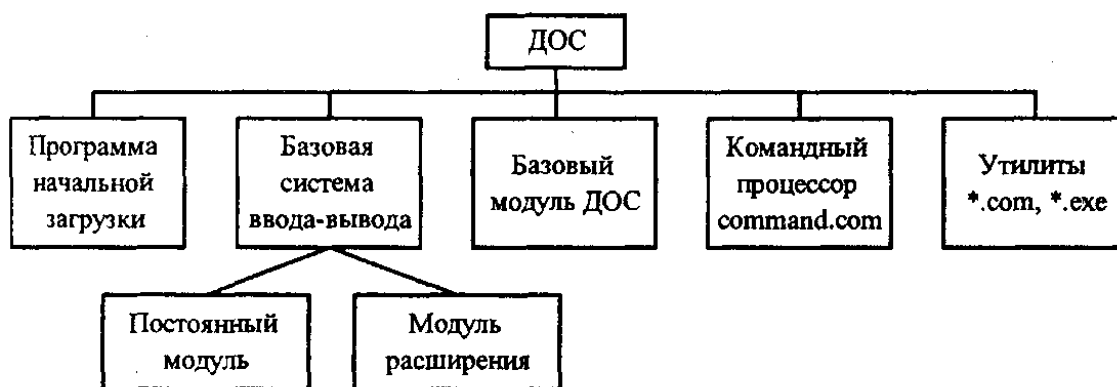


Рис. 8.2. Структура ДОС ПЭВМ

Программа начальной загрузки (Boot Record) находится в первом секторе на нулевой дорожке системного диска. Она занимает объем 512 байт. После включения компьютера и его проверки постоянный модуль *BIOS* формирует вызов данной программы и ее запуск. Назначение программы начальной загрузки – вызов модуля расширения *IO.sys* и базового модуля ДОС *MSDOS.sys*.

Базовая система ввода-вывода (BIOS) является надстройкой аппаратурной части компьютера. Постоянный модуль *BIOS* отвечает за тестирование компьютера после его включения, вызов программы начальной загрузки. Модули *BIOS* обрабатывают прерывания вычислительного процесса нижнего уровня и обслуживают стандартную периферию: дисплей, клавиатуру, принтер, дисководы.

Модуль расширения BIOS обеспечивает подключение к компьютеру дополнительных периферийных устройств, изменение некоторых параметров ДОС, замещение некоторых стандартных функций, загрузку командного процессора и его запуск.

Базовый модуль ДОС (MSDOS.sys или IBMDOS.com) отвечает за работу файловой системы, обслуживает прерывания верхнего уровня (32...63), обеспечивает информационное взаимодействие с внешними устройствами.

Командный процессор (command.com) предназначен для выполнения команд, загружаемых в командную строку ДОС. Все команды ДОС делят на внутренние и внешние. *Внутренние команды* содержатся внутри самого файла *command.com*. *Внешние команды* – это требования запуска каких-либо программ, находящихся на дисках. Кроме этого командный процессор выполняет команды файла *autoexec.bat*, если он находится на системном диске.

Файл autoexec.bat содержит список команд, выполнение которых позволяет развернуть в оперативной памяти компьютера некоторый набор вспомогательных программ или пакетов для обеспечения последующей работы пользователя.

Файл config.sys отражает специфические особенности формирования конфигурации компьютера, т.е. состава его технических и программных средств.

В связи с постоянным совершенствованием ПЭВМ и улучшением их характеристик (быстродействия и емкости памяти) все большее число ЭВМ используют более сложные интегрированные ОС типа *MS Windows 95*, *OS/2 Warp* и им подобные.

8.4. Системы автоматизации программирования

К системам (или средствам) автоматизации программирования (САП) относят языки программирования, языковые трансляторы, редакторы, средства отладки и другие вспомогательные программы.

Языки программирования служат средством передачи информации и записи текстов исходных программ. Поэтому в состав программ общего ПО они не входят.

В настоящее время известно несколько сотен языков программирования, которые используют пользователи при разработке своих заданий. Вместе с тем число активно применяемых языков программирования относительно невелико. Для каждого класса ЭВМ всегда существует несколько таких языков, ориентированных на определенные виды обработки информации, на уровень подготовки пользователей в области программирования. Различают три уровня пользователей, работающих с языковыми средствами: *пользователи-прикладники*, *системные программисты* и *инженерно-технический персонал*, обеспечивающий техническое обслуживание ЭВМ. Каждая категория пользователей использует определенный набор языков.

Важнейшие характеристики языка – трудоемкость программирования и качество получаемого программного продукта. Для языков различного уровня эти характеристики взаимосвязаны. Чем выше уровень языка (рис. 8.3), тем меньше трудоемкость программирования, но сложнее средства САП (трансляторы, средства отладки и др.), привлекаемые для получения машинных программ, тем ниже качество генерируемых программных продуктов.

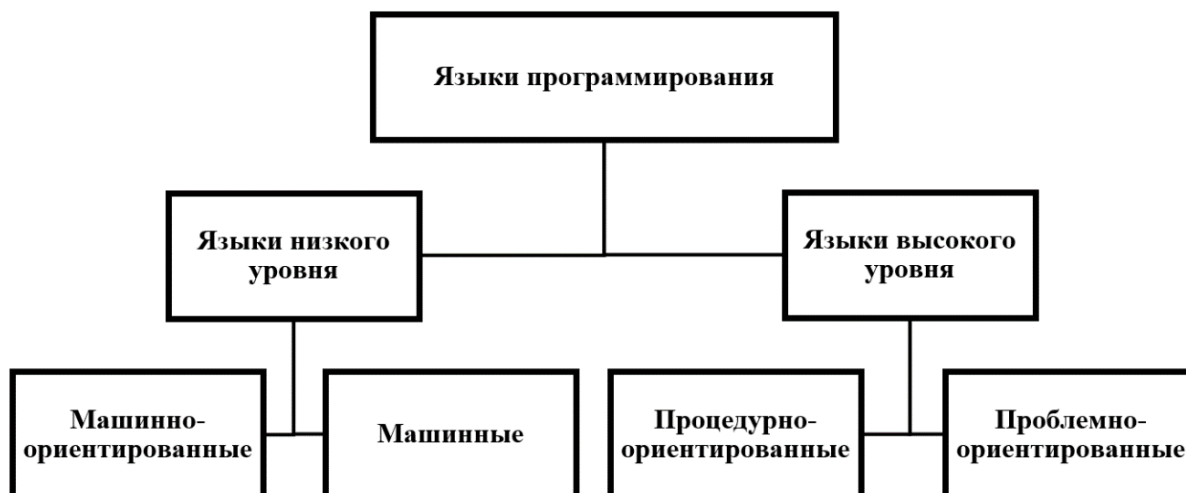


Рис. 8.3. Классификация языков программирования

Машинные языки современных ЭВМ практически не используются для программирования даже программистами-профессионалами из-за чрезмерной трудоемкости процесса разработки программ. В редких случаях их используют инженерно-технические работники вычислительных центров для проверок работы устройств и блоков ЭВМ, выяснения нестандартных, нештатных ситуаций.

Особое место имеют *машинно-ориентированные языки* (*Assembler*, автокоды, языки символического кодирования и др.). Ими часто пользуются профессиональные системные программисты при разработке компактных и быстродействующих программ общего или специального ПО.

Из *процедурно-ориентированных языков* широко известны языки *Fortran, Algol, Colob, Basic, Pascal, Ada*, СИ и др. Спектр языков этой группы очень широк, и среди них существует определенная иерархия. Считается, что язык *Basic* предназначается для начинающих програм-

мистов, язык *Pascal* – для студентов, это язык «правильного», классического программирования, язык СИ – язык квалифицированных программистов и т.д.

Все описанные выше языки программирования используют так называемые пошаговые описания алгоритмов. Именно в этом и заключается источник большой трудоемкости подготовки задач к решению. Несомненно, что для машин будущих поколений будут предложены более эффективные средства программирования. Так, например, все больше внимания уделяется разработке *проблемно-ориентированных языков* программирования (*Simula, GPSS* и др.). В данных языках имеется возможность описывать специфические алгоритмы обработки информации более крупными конструкциями. Это делает программы пользователей более наглядными, так как каждая используемая конструкция соответствует вполне определенному объекту, исследуемому пользователем.

Другая интересная тенденция – появление *непроцедурных описательных языков*. Конструкции этих языков констатируют, какой результат желателен пользователю, не указывая, каким образом это сделать. Примером такого языка служит язык ПРОЛОГ (ПРОграммирование ЛОГики), который широко используется специалистами в области искусственного интеллекта. Конструкции языка соответствуют не математическим формулам, а определяют отношения между объектами и величинами. Язык состоит только из описаний и не имеет как таковых команд-инструкций.

С усложнением структуры программ популярностью пользуется иной, нежели в структурных языках, подход к программированию. Основная часть программистов предпочитает объектно-ориентированное программирование (ООП), поскольку оно сочетает в себе лучшие идеи, воплощенные в структурном программировании, и новые концепции, позволившие оптимально организовать программу. ООП дает возможность разложить проблему на связанные между собой задачи. Каждое событие становится самостоятельным объектом, содержащим свои собственные коды и данные, которые относятся к этому объекту. Исходный код объектно-ориентированной программы из-за упрощения структуры становится более понятен и существенно короче, а программист получает возможность оперировать сложнейшими структурами данных.

Не так давно утвержден стандарт на *C++ ANSI/ISO, Java 2 (v1.2), O'Caml 2*. В 2000 г. у известного к тому моменту *Java 2 (v1.3)* возник конкурент – *C#*. Появились наиболее стабильная из существовавших версия *Perl 5.6* и получивший широкое распространение в Европе функциональный, объектно-ориентированный язык *O'Caml 3*.

К визуальным средствам разработки можно отнести такие системы проектирования, как *AutoCAD*, системы лабораторных исследований *LabView, MATLAB*, математический пакет *Maple*. Все они являются лидерами в своей области. Сюда же можно отнести семейство языков *Microsoft Visual Studio*, ставших мощнейшим инструментом в руках *Microsoft* для пропаганды миграции на платформу *Windows*, и разрозненное множество систем от *Borland* – таких сред, как *Delphi, Kylix, CBuilder* и *JBuilder*.

Будущее принадлежит кроссплатформенным инструментам разработки, к ним можно отнести *wxWindows, Qt* и несколько других пакетов.

Основные причины их популярности следующие:

- 1) низкая цена;
- 2) доступность исходных кодов;
- 3) простота программирования;
- 4) поддержка большинства современных компиляторов.

Последнее время высока популярность *WWW*-программирования, языки которого обладают рядом свойств, позволяющих использовать их на платформе, специализированной для работы в качестве сервера. Чаще всего это интерпретаторы (такие как *Perl, PHP*), языки, браузеры, поддерживаемые клиентом, – *HTML, XML, Java, JavaScript*, или специальные модули (*plug-in*), расширяющие клиента, – *Flash*.

В качестве основных тенденций в развитии современных систем программирования следует указать внедрение в них средств разработки на основе так называемых языков четвертого поколения *4GL (four generation languages)*, а также поддержки систем быстрой разработки программного обеспечения *RAD (rapid application development)*.

Языки четвертого поколения *4GL* представляют собой широкий набор средств, ориентированных на проектирование и разработку программного обеспечения. Они строятся на основе оперирования не синтаксическими структурами языка и описаниями элементов, а представляющими их графическими образами. На таком уровне проектировать

и разрабатывать прикладное программное обеспечение может пользователь, не являющийся квалифицированным программистом, зато имеющий представление о предметной области, на работу в которой ориентирована прикладная программа. Языки четвертого поколения – это следующий (четвертый по счету) этап в развитии систем программирования.

Описание программы, построенное на основе языков *4GL*, транслируется затем в исходный текст и файл описания ресурсов интерфейса, представляющие собой обычный текст на соответствующем входном языке высокого уровня. С этим текстом уже может работать профессиональный программист-разработчик – корректировать и дополнять его необходимыми функциями. Такой подход позволяет разделить работу проектировщика, ответственного за общую концепцию всего проекта создаваемой системы, дизайнера, отвечающего за внешний вид интерфейса пользователя, и профессионального программиста, отвечающего непосредственно за разработку исходного кода создаваемого программного обеспечения.

В целом языки четвертого поколения решают уже более широкий класс задач, чем традиционные системы программирования. Они составляют часть средств автоматизированного проектирования и разработки программного обеспечения, поддерживающих все этапы жизненного цикла *CASE*-систем.

В состав САП включаются также *языковые трансляторы* для всех языков, которые используют пользователи при разработке своих программ. В зависимости от специфики вычислительного центра и контингента пользователей их состав формируется эмпирически. Обычно же он включает трансляторы процедурно-ориентированных языков высокого уровня (*Pascal*, *Basic*, Си) и машинно-ориентированных языков (*Assembler*).

Различают трансляторы двух типов: интерпретаторы и компиляторы.

Трансляторы-интерпретаторы предназначаются для последовательного пооператорного преобразования каждого предложения исходного модуля программы в блок машинных команд с одновременным их выполнением. Машинная программа в полном объеме при этом не создается, решение задач пользователей происходит замедленными темпами. Этот вид трансляции рекомендуется использовать при отладке новых программных продуктов.

Трансляторы-компиляторы, напротив, предназначены для формирования полного загрузочного модуля по исходным программам пользователя. Это позволяет отделить полученный программный продукт от среды его разработки и в последующем использовать автономно.

Из *системных обслуживающих программ*, широко используемых при подготовке вычислений, следует выделить *редактор*, *загрузчик*, *библиотекарь*, *средства отладки* и другие вспомогательные программы. Назначение программ вытекает из их названий.

Программы пользователей после обработки их транслятором (трансляторами) представляются в виде набора программных блоков, имеющих промежуточный формат, общий для всех трансляторов. Специфика исходных языков программирования при этом теряется. Объединение программных блоков в единую программу выполняет *редактор*. В зависимости от того, в какой стадии подготовки к решению находятся программы абонентов, они могут размещаться в различных библиотеках. Управляет размещением программ, последующей идентификацией и выборкой *библиотекарь*. Вызов готовых к решению программ в оперативную память, активизацию их с учетом места размещения выполняет *загрузчик*. *Средства отладки* обеспечивают проверку заданий пользователей, поиск в них различного рода ошибок, вывод на печать запрашиваемой отладочной информации, распечатку содержимого зон оперативной памяти, выдачу различных управляющих блоков и таблиц и т.п.

Вспомогательные программы (утилиты) служат для перемещения информации с одного носителя на другой, разметки накопителей, редактирования информации в наборах данных, сбора информации об ошибках.

8.5. Комплекс программ технического обслуживания

Особенности состава технических средств ЭВМ учитываются КПО. Этот комплекс включает в свой состав наладочные, проверочные и диагностические тест-программы.

Наладочные программы обеспечивают автономную настройку и проверку отдельных устройств ЭВМ. Обычно они функционально независимы от программ ОС. *Проверочные тест-программы* предназна-

чены для периодически проводимых проверок правильности функционирования устройств, например, после включения их в работу. *Диагностические программы* используются в тех случаях, когда необходимо классифицировать отказ оборудования и локализовать место неисправности. Инициирование работы данных программ осуществляется обычно модулями ОС после фиксации сбоев и отказов аппаратуры контроля.

У *IBM PC* эти средства имеют своеобразную структурную и функциональную организацию, часть средств записана в ПЗУ компьютера. При каждом включении ПЭВМ и перезагрузках производится ее предварительная проверка путем выполнения тестовой программы *POST (Power On Set Test)*, состоящей из более десятка отдельных программных фрагментов.

Последовательность проверок заключается в следующем. Вначале проверяется работоспособность системного блока. Для этого все регистры машины "сбрасываются в нуль", и производится их последовательная проверка путем занесения отдельных констант, выполнения над ними простейших операций и сравнения результатов с эталонными значениями. После этого проверяются ячейки оперативной памяти, клавиатура, накопители на дисках, дисплей и др. В случае каких-либо ошибок на каждом шаге проверки формируются определенные звуковые сигналы, сопровождаемые соответствующими сообщениями на экране дисплея.

Кроме встроенных средств контроля в ПО ПЭВМ включаются и автономные средства контроля и диагностики. Количество подобных комплектов программ достаточно велико, и каждый из них позволяет детализировать системную информацию – определить полную конфигурацию ПК и характеристики отдельных ее частей (тип процессора, наличие сопроцессора, тип материнской платы, типы используемых дисков, объем оперативной памяти и ее распределение, подключение дополнительной периферии и т.д.). Примеры таких систем – интегрированные тестовые программы *Checkt* и *WinCheckt*.

Помимо контроля работоспособности они могут отразить, насколько эффективно используются ресурсы, и осуществить их перераспределение.

Все пользователи стараются пополнить программное обеспечение ПЭВМ вспомогательными системными *программами-утилитами*. Эти программы напрямую в вычислительном процессе не используются, а обеспечивают необходимый и разнообразный сервис при подготовке заданий пользователями. Часть таких программ может быть объединена в пакеты. Широкое распространение получили такие пакеты, как *Norton Utilities*, *PC Tool Deluxe* и др. Примерами подобных программ могут быть: программы-архиваторы, антивирусные программы, программы обслуживания дисков (оптимизация дисков, сжатие информации на дисках, определение состояния диска) и др.

8.6. Специальное программное обеспечение. Пакеты программ

В ПО ЭВМ, как ранее упоминалось, имеются две группы пакетов программ: ППП и ППос, их довольно часто сливают в единые интегрированные пакеты. Например, операционная среда *Windows* подключает и пакеты *MS Office*, объединяющие программы для работы экономиста-делопроизводителя.

ППП – это комплекс программ, предназначенных для решения определенного класса задач пользователей. Все чаще к ППП относят наряду с комплексом готовых программ и программную среду, оболочку, в которой создаются пользовательские программы. Наиболее широкое распространение получили следующие прикладные системы, обеспечивающие различные виды работ пользователей:

- системы обработки текстов (текстовые редакторы);
- системы обработки электронных таблиц;
- системы управления базами данных;
- системы деловой графики;
- коммуникационные системы, САПР и т.д.

Первые три системы предназначаются для работы с документами. Практически каждый документ содержит текстовую часть. С текстом работают *текстовые процессоры* (*Word Processors – WP*). В настоящее время известны десятки и сотни *WP*, различающиеся направленностью, возможностями, сервисом, качеством формируемых документов. Независимо от назначения каждый редактор должен обеспечивать выполнение следующих процедур подготовки текста:

- набор текста;
- редактирование текста (форматирование, поиск и замена данных, работа с фрагментами текста, подготовка к печати и др.);
- печать документов;
- ведение архивов документов.

Самые простые редакторы встраиваются в многие пакеты, включая ППП и ППос. Они имеют достаточно скромные характеристики. Примерами подобных *WP* являются: встроенный редактор командной строки *DOS*, редактор *Edit* пакета *Total Commander* и др.

Другие редакторы – *редакторы широкого назначения*, обычно используются автономно. Они получили большую признательность пользователей и используются повсеместно при отработке документов различной сложности, от простейших справок до создания издательского макета фундаментальных книг (*Word, ChiWriter, MultiEdit* и др.). Последние версии редакторов типа *Word* от *Windows* предоставляют пользователям возможности настольной издательской системы.

Наиболее сложными редакторами являются *WP мощных издательских систем*, предназначенных для оформления и полной подготовки к типографскому изданию книг, журналов, буклетов (*Aldus PageMaker, Ventura Publisher*). Они позволяют включать в текст фотографии, иллюстрации, графики, диаграммы, использовать различные шрифты, менять параметры текста, осуществлять перемещение фрагментов, изменять оформление документа, автоматизировать его верстку.

Системы обработки электронных таблиц, или *табличные процессоры*, предназначены для работы с фактографическими документами. Этот вид документа представляет собой двухмерные таблицы, как правило, заранее определенной формы, каждая клетка которой содержит значение некоторой характеристики объекта. Подобные документы являются наиболее распространенными в деятельности различных отделов, служб, предприятий и т.п. Примерами этих документов могут служить бухгалтерские ведомости, отчеты, планы, списки и прочее. Такие документы представляются в памяти ЭВМ в виде электронных таблиц.

В деятельности многих фирм широко используются такие пакеты, как *SuperCalc, Lotus 1-2-3, Quattro Pro, Excel*.

Области применения табличных процессоров достаточно широки. Наиболее простые позволяют обрабатывать табличные данные по формулам, что очень важно при решении задач статистического анализа и при обработке экспериментальных данных. В более сложных процедурах моделирования и прогнозирования требуется решать задачи экстраполяции и интерполяции.

Еще одной группой ППП являются *системы управления базами данных* (СУБД). *База данных* (БД) – это совокупность взаимосвязанных данных, хранящихся совместно в памяти ЭВМ. Каждая БД состоит из *записей*. Запись образует подмножество данных, служащих для описания единичного объекта. Например, фамилия, имя, отчество, год рождения, адрес, место работы, номер телефона могут составлять одну запись и характеризовать одного человека. Информационный массив может содержать записи по отдельным цехам, службам, отделам всего предприятия.

СУБД автоматизирует работу пользователей с хранящимися данными. Количество информационных массивов в БД и их объем зависят от сложности создаваемой системы. Ядро БД составляет информация, наиболее часто используемая в процессах управления. Согласно принципу В. Парето (итальянский экономист XIX в.) 20 % всей информации обеспечивают более 80 % всех задач управления. Эта часть в первую очередь и подлежит автоматизации.

Одно из основных назначений СУБД – *автоматизация документооборота*. На основе хранящейся информации можно автоматически формировать любые стандартные документы. СУБД обеспечивает ввод, поиск, сортировку данных, составление отчетов. Широко используются следующие СУБД: *Dbase IV (V), Clipper, Access Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual Fox-Pro, Microsoft Visual Basic* и др. Все они, в свою очередь, состоят из языковых и программных средств. Различие между ними состоит в предлагаемом сервисе и удобствах работы.

В современных СУБД применяется клиент-серверный подход. В рамках этого подхода СУБД представляет собой сервер (поставщик услуг), который принимает сетевые соединения, исходящие от программ-клиентов (потребителей услуг). Клиентские программы могут

работать как на том же компьютере, что и программа-сервер, так и на других компьютерах. Как только соединение установлено, клиент может отправлять запросы в сторону сервера и получать от него ответы. Таким образом, СУБД отвечает только за обработку запросов и хранение данных. Всю интерактивную часть (то есть связанную с взаимодействием с потребителем) берут на себя клиентские программы. Примером таких СУБД служат коммерческая система *Oracle* производства одноимённой фирмы и свободно распространяемые *MySQL* и *PostgreSQL*.

Графические редакторы (ГР) позволяют создавать и редактировать на экране компьютера различные рисунки, диаграммы, картинки. Своеобразие их построения заключается в том, что на экране информация представляется в виде точек, линий, окружностей, прямоугольников, кривых. Элементы рисунков могут использовать различное сочетание цветов, шрифтов, форматов. Допускается работа с фрагментами рисунков. Некоторые ППП имеют собственные встроенные графические редакторы. Они ориентированы на специфические режимы работы (графики, диаграммы). Некоторые редакторы допускают автономную работу, что очень важно в системах деловой и научной графики. Например, графический редактор *Paintbrush for Windows* позволяет создавать фрагменты изображений и включать их в другие программные продукты.

По мере накопления опыта разработки и применения ППП пользователи стали переходить к эксплуатации *интегрированных систем*, объединяющих наиболее часто используемые прикладные системы и пакеты. Интегрированные пакеты программ можно рассматривать как дальнейшую надстройку ОС, так как в них аккумулируются средства, определяющие специфику работы конкретного пользователя (работа в многомашинных и многопроцессорных системах, работа в сети ЭВМ, реализация определенных режимов и т.д.).

Сочетание различных видов обработки в рамках единой операционной среды создает дополнительные удобства пользователям. Упрощение общения достигается путем разработки *дружественного ПО* путем подсказок, инструкций, предоставления вариантов действий и т.д. Фирмы-разработчики таких пакетов стараются сохранить в них единые принципы представления информации, управления и работы.

Вопросы к компьютерному тестированию

1. Приведите названия двух групп, на которые разделяют программное обеспечение ЭВМ.
2. Что включают в состав общего программного обеспечения? Приведите аббревиатуры ОС.
3. Как называют носитель с соответствующим набором программ дисковой операционной системы?
4. Какая из известных дисковых ОС имеет развитые средства защиты информации и разграничение доступа?
5. Какая из известных дисковых ОС позволяет более полно использовать возможности самых мощных микропроцессоров?
6. Какая из известных дисковых ОС ориентирована на работу ЭВМ в сети?
7. Какие из представленных в главе программ образуют ядро ОС?
8. Какие из программ после включения компьютера и его проверки вызываются программой начальной загрузки?
9. Какая из представленных в главе программ осуществляет вызов программы начальной загрузки?
10. Какие из представленных в главе программ обеспечивают информационное взаимодействие с периферией?
11. Какая из представленных в главе программ предназначена для выполнения загружаемых команд?
12. Какая из перечисленных программ формирует набор вспомогательных пакетов для обеспечения последующей работы пользователя?
13. Какая из представленных в главе программ отражает состав технических и программных средств компьютера?
14. Какой из представленных в главе языков относят к машинным языкам?
15. Какой из представленных в главе языков относят к машинно-ориентированным языкам?
16. Какие из представленных в главе языков относят к процедурно-ориентированным языкам?

17. Какой из представленных в главе языков относят к проблемно-ориентированным языкам?
18. В каких языках имеется возможность описывать специфические алгоритмы обработки информации крупными конструкциями?
19. Какой из представленных в главе языков относят к непроцедурным описательным языкам?
20. В каком языке программирования конструкции языка соответствуют не математическим формулам, а определяют отношения между объектами и величинами?
21. В каких языках программирования оперируют не синтаксическими структурами языка и описаниями элементов, а представляющими их графическими образами?
22. Что относят к языковым трансляторам?
23. Приведите название транслятора, предназначенного для последовательного пооператорного преобразования каждого предложения исходного модуля программы.
24. Приведите название транслятора, предназначенного для формирования полного загрузочного модуля.
25. Какие тест-программы включает в свой состав комплекс программ технического обслуживания (КПТО)?
26. Как называют программы, напрямую в вычислительном процессе не использующиеся, а обеспечивающие необходимый и разнообразный сервис при подготовке заданий пользователями?
27. На какие группы разделяют пакеты программ специального программного обеспечения?
28. Приведите один из примеров редактора мощных издательских систем.
29. Каково основное назначение СУБД?
30. Приведите один из примеров коммерческой или свободно распространяемой СУБД с клиент-серверным подходом.

Глава 9

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ



Рассматриваемые вопросы:

- 9.1. Общие положения и классификация вычислительных систем.
- 9.2. Архитектура вычислительных систем.
- 9.3. Кластеризация вычислительных систем.
- 9.4. Организация функционирования вычислительных систем.
- 9.5. Методы комплексирования вычислительных систем.
- 9.6. Суперкомпьютеры и особенности их архитектуры.

9.1. Общие положения и классификация вычислительных систем

Более чем за полувековую историю развития сменилось четыре поколения вычислительных машин. Эти поколения были машинами с фон-неймановской архитектурой.

Основная особенность этой архитектуры:

- 1) хранение программ и данных в одной памяти;
- 2) линейная организация и упорядоченность ячеек памяти;
- 3) последовательное выполнение команд;
- 4) отсутствие различий между представлением команд и данных в машинной памяти (все в двоичном коде).

Это все постепенно привело к кризису, основными причинами которого стали:

- резкое увеличение требований к надежности вычислительного процесса, поскольку системы с последовательной организацией мало-надежны;
- увеличение сложности программного обеспечения;

- увеличение сложности задач, приводящее к тому, что время решения задач становится недопустимо большим (поскольку машины фон-неймановской архитектуры были программно-управляемыми);
- намечающаяся тенденция к переходу от обработки данных к обработке знаний.

В связи с этим дальнейшее развитие вычислительной техники было напрямую связано с переходом к параллельным вычислениям, с идеями построения многопроцессорных систем и сетей, объединяющих большое количество отдельных процессоров и (или) ЭВМ. Машины 5-го поколения – машины обработки знаний.

Отражая эти новшества, и появился термин “вычислительная система”, который не имеет единого толкования в литературе, его иногда даже используют применительно к однопроцессорным ЭВМ. Однако общим здесь является подчеркивание возможности построения параллельных ветвей в вычислениях, что не предусматривалось классической структурой ЭВМ.

Под *вычислительной системой* понимают совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенную для сбора, хранения, обработки и распределения информации.

Отличительная особенность ВС по отношению к ЭВМ – наличие в них нескольких вычислителей, реализующих параллельную обработку. Создание ВС преследует определенные основные цели:

- повышение производительности системы за счет ускорения процессов обработки данных;
- повышение надежности и достоверности вычислений;
- предоставление пользователям дополнительных сервисных услуг и т.д.

Параллелизм в вычислениях в значительной степени усложняет управление вычислительным процессом, использование технических и программных ресурсов. Эти функции выполняет операционная система ВС [9].

Самыми важными предпосылками появления и развития вычислительных систем служат экономические факторы. Анализ характеристик ЭВМ различных поколений показал, что в пределах интервала времени, характеризующегося относительной стабильностью элемент-

ной базы, связь стоимости и производительности ЭВМ выражается квадратичной зависимостью – “законом Гроша”:

$$C_{\text{ЭВМ}} = K_1 P_{\text{ЭВМ}}^2.$$

Построение же вычислительных систем позволяет существенно сократить затраты, так как для них существует линейная формула

$$C_{\text{ВС}} = K_2 \sum_{i=1}^n P_i,$$

где $C_{\text{ЭВМ}}$, $C_{\text{ВС}}$ – соответственно стоимость ЭВМ и ВС, K_1 и K_2 – коэффициенты пропорциональности, зависящие от технического уровня развития вычислительной техники, $P_{\text{ЭВМ}}$, P_i – производительность ЭВМ и i -го из n комплектующих вычислителей (ЭВМ или процессоров).

На рис. 9.1 представлены графики изменения стоимости вычислений для ЭВМ и ВС в зависимости от производительности. Для каждого поколения ЭВМ и ВС существует критический порог сложности решаемых задач $P_{\text{кр}}$, после которого применение автономных ЭВМ становится экономически невыгодным, неэффективным. Критический порог определяется точкой пересечения двух приведенных зависимостей.

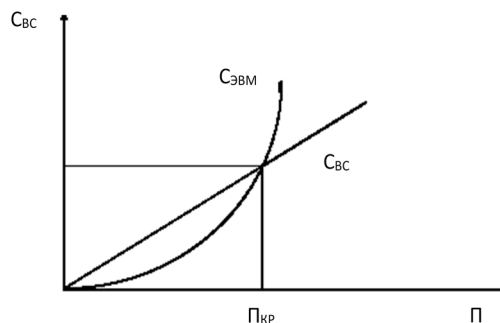


Рис. 9.1. Зависимость стоимость – производительность

В настоящее время накоплен большой практический опыт в разработке и использовании ВС самого разнообразного применения. Эти системы очень сильно отличаются друг от друга своими возможностями и характеристиками.

Существует большое количество признаков, по которым классифицируют вычислительные системы.

По назначению вычислительные системы делят на *универсальные* и *специализированные*. Специализированные системы ориентированы на решение узкого класса задач в отличие от универсальных, предназначенных для широкого спектра задач.

По типу вычислительные системы различаются на *многомашинные* и *многопроцессорные ВС*. Многомашинные вычислительные системы (ММС) появились раньше, чем многопроцессорные. Основные отличия ММС заключаются в организации связей и обмена информацией между ЭВМ комплекса. Каждая из них сохраняет возможность автоном-

ной работы и управляется собственной ОС. Любая другая подключаемая к ней ЭВМ рассматривается как периферийное специальное устройство.

Многопроцессорные системы (МПС) строятся при комплексировании (от 2 до 10) процессоров. В качестве общего ресурса они имеют общую оперативную память (ООП). Параллельная работа процессоров с ООП обеспечивается под управлением единой общей операционной системы. По сравнению с ММС здесь достигается наивысшая оперативность взаимодействия процессоров-вычислителей.

По типу ЭВМ или процессоров, используемых для построения ВС, различают однородные и неоднородные системы. Однородные системы предполагают комплексирование однотипных ЭВМ (процессоров), неоднородные – разнотипных. В однородных системах значительно упрощаются разработка и обслуживание технических и программных средств, обеспечивается возможность унификации, упрощаются обслуживание, модернизация и развитие систем.

По степени территориальной разобщенности вычислительных модулей ВС делятся на системы *совмещенного (сосредоточенного) и распределенного (разобщенного)* типов. Обычно такое деление касается только ММС. Многопроцессорные системы относятся к системам совмещенного типа. Более того, учитывая успехи микроэлектроники, это совмещение может быть очень глубоким. При появлении новых СБИС появляется возможность иметь в одном кристалле несколько параллельно работающих процессоров.

По методам управления элементами различают *централизованные, децентрализованные и со смешанным управлением* ВС.

- *В централизованных ВС* за управление отвечает главная или диспетчерская ЭВМ (процессор). Ее задача – распределение нагрузки между элементами, выделение ресурсов, контроль состояния ресурсов, координация взаимодействия. Централизованные системы имеют более простые ОС.

- *В децентрализованных системах* функции управления распределены между ее элементами. Каждая ЭВМ (процессор) системы сохраняет известную автономию, а необходимое взаимодействие между элементами устанавливается по специальным наборам сигналов. С развитием ВС и, в частности, сетей ЭВМ, интерес к децентрализованным системам постоянно растет.

- В системах со смешанным управлением совмещаются процедуры централизованного и децентрализованного управления. Перераспределение функций осуществляется во время вычислительного процесса, исходя из сложившейся ситуации.

По принципу закрепления вычислительных функций за отдельными ЭВМ (процессорами) различают системы с жестким и плавающим закреплением функций. В зависимости от типа ВС следует решать задачи статического или динамического размещения программных модулей и массивов данных, обеспечивая необходимую гибкость системы и надежность ее функционирования.

По режиму работы ВС различают системы, работающие в оперативном и неоперативном временных режимах. Первые, как правило, используют режим реального масштаба времени. Этот режим характеризуется жесткими ограничениями на время решения задач в системе и предполагает высокую степень автоматизации процедур ввода/вывода и обработки данных.

9.2. Архитектура вычислительных систем

Архитектура ВС – это совокупность характеристик и параметров, определяющих функционально-логическую и структурную организацию системы.

Основное отличие ВС от отдельных ЭВМ – наличие в их структурах нескольких вычислителей (ЭВМ или процессоров), поэтому они способны выполнять параллельные вычисления.

Поскольку ВС относятся к параллельным системам, то и рассмотрим классификацию архитектур с этой точки зрения. Эта классификация архитектур была предложена Флинном (*M. Flynn*) в начале 60-х гг. В ее основу заложено два возможных вида параллелизма:

- независимость потоков заданий (команд), существующих в системе;
- независимость (несвязность) данных, обрабатываемых в каждом потоке.

Согласно данной классификации существуют четыре основных архитектуры ВС:

1. ОКОД (*SISD*) (*Single Instruction Single Data*) – один поток команд, один поток данных (рис. 9.2).

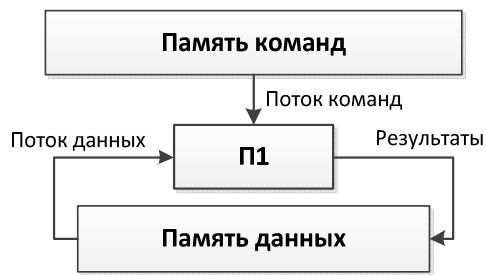


Рис. 9.2. Архитектура ОКОД

Это все машины фон-неймановской архитектуры. Главная проблема архитектуры – наличие побочного эффекта – пересылки «процессор – память».

Еще одна из разновидностей этой архитектуры – *CISC* (*Complete Instruction Set Command*) – система с полным набором команд, т.е. есть команды «на все случаи жизни».

Команды в полном наборе – команды переменной длины. От этого полного набора постоянно используются около 30 % команд, а все остальное – балласт. Отношение времени обращения к памяти к времени вычисления равно 5 : 1.

Другая разновидность – архитектура *RISC* (*Reduced Instruction Set Command*) – с сокращенным набором команд. В качестве основных команд оставили только самые короткие и наиболее часто используемые, а длинные команды реализовывались через подпрограммы. В системах типа *RISC* заметно увеличился объем сверхоперативной памяти. Основная пересылка «регистр – регистр». В данном случае отношение времени обращения к памяти к времени вычисления

равно 2 : 1.

Еще одна разновидность *VLIW* (*Very Large Instruction Word*) – системы с большим командным словом. Командное слово машины – набор элементарных инструкций. Обычная длина слова от 32 до 512 разрядов. Эти слова могут обрабатываться параллельно во времени.



Рис. 9.3. Архитектура ОКМД

2. ОКМД (*SIMD*) (*Single Instruction Multiple Data*) – один поток команд, много потоков данных (рис. 9.3). Системы этого типа обычно строятся как однородные, то есть процессорные элементы, входящие в систему, идентичны, и все они управ-

ляются одной и той же последовательностью команд. Однако каждый процессор обрабатывает свой поток данных. Применяются для скалярного, векторного, матричного произведения. Увеличение быстродействия при увеличении объема аппаратных заданий.

3. МКОД (MISD) (Multiple Instruction Single Data) – много потоков команд, один поток данных (рис. 9.4). Теоретически структура возможна, но практически задач с такой структурой очень мало. Одна из разновидностей – конвейерная архитектура. В классической системе МКОД предполагается, что каждый следующий процесс начинает обрабатывать данные только после того, как предшествующий закончит свои операции. В конвейерных системах операции соседних процессоров частично перекрываются во времени. За счет этого перекрытия и достигается производительность системы.



Рис. 9.4. Архитектура МКОД

4. МКМД (MIMD) (Multiple Instruction Multiple Data) – множественный поток инструкций, множественный поток данных MIMD (рис. 9.5). Архитектура МКМД предполагает, что все процессоры системы работают по своим программам с собственным потоком команд. Иногда говорят, что это системы с массовым параллелизмом. Возможно построение систем с десятками, сотнями и даже тысячами процессорных элементов с размещением их в непосредственной близости друг от друга. Если каждый процессор системы имеет собственную память, то он также будет сохранять известную автономию в вычислениях. Считается, что именно такие системы займут доминирующее положение в мире суперкомпьютеров в ближайшие десять-пятнадцать лет.

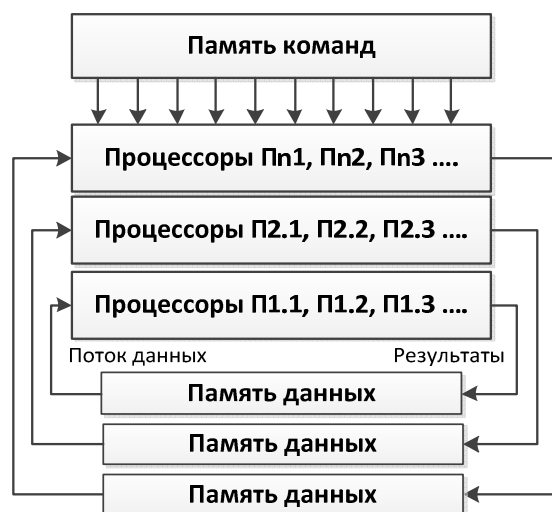


Рис. 9.5. Архитектура МКМД

К структурам типа МКМД относятся SMP-системы (Shared Memory-multi Processing). На локальную общую шину подключаются

порядка четырех процессоров, которые взаимодействуют внутри локальной шины, и только, если это необходимо, выходят на глобальную. Система *SMP* не является системой с массовым параллелизмом.

Системы типа МКМД могут быть как многомашинные, так и многопроцессорные. Первые сейчас относят к вычислительным системам, поэтому, когда речь идет о системах типа МКМД, по умолчанию предполагают, что это многопроцессорные системы. К системам с массовым параллелизмом относят системы с матричной организацией. Под процессорными элементами понимаются все элементы системы, необходимые для функционирования (процессор, память, устройства ввода/вывода).

9.3. Кластеризация вычислительных систем

Вычислительные системы как мощные средства обработки заданий пользователей широко используются не только автономно, но и в сетях ЭВМ в качестве серверов.

С увеличением размеров сетей и их развитием возрастают плотности информационных потоков, нагрузка на средства доступа к сетевым ресурсам и на средства обработки заданий. Круг задач, решаемый серверами, постоянно расширяется, становится многообразным и сложным. Чем выше ранг сети, тем более специализированными они становятся. Администраторы сетей должны постоянно наращивать их мощь и количество, оптимизируя характеристики сети под возрастающие запросы пользователей.

Одно из перспективных направлений здесь – *кластеризация*, т.е. технология, с помощью которой несколько серверов, являющихся вычислительными системами, объединяются в единую систему более высокого ранга для повышения эффективности функционирования системы в целом.

Целями построения кластеров могут служить:

- улучшение масштабируемости;
- повышение надежности и готовности системы в целом;
- увеличение суммарной производительности;
- эффективное перераспределение нагрузок между компьютерами кластера;
- эффективное управление и контроль работы системы и т.п.

Улучшение масштабируемости, или способности к наращиванию мощности, предусматривает, что все элементы кластера имеют аппаратную, программную и информационную совместимость. Изменение оборудования в идеальном кластере должно обеспечивать соответствующее изменение значений основных характеристик, т.е. добавление новых процессоров, дисковых систем должно сопровождаться пропорциональным ростом производительности, надежности и т.п.

Следующая важная цель создания кластера – *повышение надежности* и готовности системы в целом. Эти качества обеспечиваются избыточностью, изначально заложенной в кластеры. Основой этого служит возможность каждого сервера кластера работать автономно, но в любой момент он может переключиться на выполнение работ другого сервера в случае его отказа.

Коэффициент готовности систем рассчитывают по формуле

$$K_r = T_p / (T_p + T_o),$$

где T_p – полезное время работы системы; T_o – время отказа и восстановления системы, в течение которого она не могла выполнять свои функции.

Большинство современных серверов имеет 99-процентную готовность. Это означает, что около четырех дней в году они не работают.

Подчеркнем, что готовность 99,9 %, достигаемая обычно спаркой серверов – основного и резервного, означает годовой простой около 500 мин, 99,999 % – 5 мин, а 99,9999 % – 30 с.

Появление критически важных приложений в областях бизнеса, финансов, телекоммуникаций, здравоохранения и других требует обеспечения коэффициента готовности не менее чем "заветные пять девяток" и даже выше.

Повышение суммарной производительности кластера, объединяющего несколько серверов, обычно не является самоцелью, а обеспечивается автоматически. Каждый сервер кластера – достаточно мощная вычислительная система, рассчитанная на выполнение всех необходимых функций в части управления соответствующими сетевыми ресурсами.

Распределенные вычисления приобретают все большее значение с развитием сетей. При этом многие компьютеры, в том числе и серверы, могут иметь не очень большую нагрузку. Для выполнения каких-либо трудоемких вычислений можно использовать свободные ресурсы

домашних компьютеров, рабочих станций локальных вычислительных сетей, да и самих серверов. При этом стоимость создания подобных вычислительных кластеров очень мала, так как все их составные части работают в сети и только при необходимости образуют виртуальный (временный) вычислительный комплекс.

Работа кластера под управлением единой операционной системы позволяет оперативно контролировать процесс вычислений и эффективно перераспределять нагрузки на компьютеры кластера.

Эффективное управление и контроль работы системы – возможность работы отдельно с каждым узлом, вручную или программно отключать его для модернизации или ремонта с последующим возвращением в работающий кластер. Эти операции скрыты от пользователей, они просто не замечают их. Кластерное ПО, интегрированное в операционные системы серверов, позволяет работать с узлами как с единым пулом ресурсов (*Single System Image – SSI*), внося необходимые общие изменения с помощью одной операции для всех узлов.

Кластеры объединяют несколько серверов под единым управлением.

Все новые серверы, как правило, являются многопроцессорными и относятся к *SMP*-структурам, что обеспечивает многоступенчатую возможность переключения нагрузки отказавшего элемента как внутри кластера, так и внутри сервера.

Большой интерес к построению кластеров стала проявлять фирма *Microsoft*. В связи с широкой популярностью операционной системы *Windows NT*, предназначенной для управления сетями крупных предприятий, появились различные варианты кластерного обеспечения. Сама фирма *Microsoft* предлагает бесплатную версию своего кластерного ПО, встроенного в *Windows NT* и поддерживающего *Microsoft Cluster Server (MSCS)*. Этот кластерный продукт, известный под названием *Wolfpack* ("волчья стая"), сильно прогрессирует. В настоящее время он обеспечивает разделение нагрузки между двумя узлами-серверами и то только путем замены одного сервера другим, а не путем ее перераспределения.

Унификация инженерно-технических решений предполагает соответственно и стандартизацию аппаратных и программных процедур обмена данными между серверами. Для передачи управляющей информации в кластере используются специальные магистрали, имеющие

более высокие скорости обмена данными. В качестве такого стандарта предлагается интеллектуальный ввод/вывод (*Intelligent Input/Output – I2O*). Спецификация *I2O* определяет унифицированный интерфейс между операционной системой и устройствами ввода/вывода, освобождая процессоры и их системные шины от обслуживания периферии.

Как и у любой новой технологии, у кластеризации имеются свои недостатки:

- задержки разработки и принятия общих стандартов;
- большая доля нестандартных и закрытых разработок различных фирм, затрудняющих их совместное использование;
- трудности управления одновременным доступом к файлам;
- сложности с управлением конфигурацией, настройкой, развертыванием, оповещениями серверов о сбоях и т.п.

9.4. Организация функционирования вычислительных систем

Организация функционирования ВС зависит от их типа.

1. *Многомашинальная система* – ее функционирование можно описать фразой «*Master-Slave*» – в каждый момент времени если выбирается одна машина из системы, то все остальные машины системы рассматриваются как «интеллектуальная» периферия этой машины. Частный случай этой организации – наличие одной главной машины, за которой жестко закреплена центральная функция.

2. *Многопроцессорная система*. Есть несколько вариантов таких систем, так как в этих жесткосвязанных системах имеется одна общая достаточно сложная ОС:

- 1-й вариант: схема «ведущий – ведомый» («главный – подчиненный»). В зависимости от характера системы эти цепочки могут быть жестко фиксированными либо идет реконфигурация системы, и в каждый момент времени создаются новые пары «ведущий – ведомый»;
- 2-й вариант: симметричная и однородная обработка во всех процессорах системы. Одно из достоинств и преимуществ – наличие высокой надежности вычисления. Для повышения надежности вычисления в многопроцессорных системах одну и ту же задачу можно запустить параллельно и синхронно по нескольким процессам, а затем сравнить полученные результаты;

- 3-й вариант: когда идет независимая, параллельная во времени работа всех процессоров, и в этом случае процессоры решают свои конкретные задачи, не зависящие от задач других процессов, и где-то в конце системы на выходе эти результаты каким-то образом объединяются.

9.5. Методы комплексирования вычислительных систем

Для того чтобы обеспечить вычислительным системам должную организацию, нужно создать комплексы вычислительных средств, а для этого необходимо обеспечить совместимость этих вычислительных средств.

Различают три уровня совместимости:

- 1) информационная совместимость;
- 2) программная совместимость;
- 3) аппаратная совместимость.

Информационная совместимость означает, что информация во всех составляющих системы должна интерпретироваться одинаковым образом (т.е. быть одинакового типа).

Программная совместимость означает то, что программы, работающие на одних компьютерах, должны работать на всех машинах системы. В многомашинных системах это требование может быть менее жестким. Для многопроцессорных систем оно должно выполняться, так как там одна ОС для всех процессоров. В многомашинных системах если предполагается передача данных с одной машины на другую, то совместимость должна быть полная, но если идет просто обмен данными, то здесь достаточно совместимости форматов.

Аппаратная совместимость – это совместимость по техническим характеристикам аппаратных средств либо вычислительных машин, либо процессоров, входящих в систему. Теоретически она предполагает, что при правильном соединении соответствующих кабелей и разъемов система должна сразу работать. Для многомашинных систем требование аппаратной совместимости достаточно мягкое, так как всегда можно поставить согласующее устройство и это требование обеспечить. Как правило, вычислительные машины связаны между собой в си-

стеме достаточно протяженными линиями связи, т.е. без устройств согласования принципиально не обойтись. В многопроцессорных системах упомянутое требование абсолютно жестко должно соблюдаться.

В зависимости от условия задачи используют следующие методы комплексирования:

- 1) метод прямого управления, либо комплексирование «процессор – процессор»;
- 2) комплексирование на уровне общей оперативной памяти;
- 3) комплексирование на уровне каналов ввода/вывода;
- 4) комплексирование на уровне внешних устройств либо на уровне управления ими;
- 5) комплексирование на уровне общих внешних устройств.

Комплексирование «процессор – процессор»

В этом случае процессоры системы напрямую взаимодействуют между собой через интерфейс прямого управления (ИПУ) (рис. 9.6). Это самая простая схема комплексирования и самая неудобная, поскольку:

- она очень неэффективна с точки зрения использования ресурсов процессора. В этом случае процессор непосредственно участвует в организации обмена, следовательно, вычислительные возможности процессора используются очень плохо (так как он должен считать, а не заниматься обменом);

- все принципы совместимости необходимо выполнить в полном объеме, поэтому процессоры по производительности должны быть жестко согласованы между собой ИПУ.

Но процессоры в системе могут работать с различной скоростью, и если, например, первый процессор выдает данные со скоростью большей, чем второй, то последний будет неспособен принимать данные. А если наоборот, то первый не до конца использует свои ресурсы. Поэтому ИПУ в первом случае накапливает данные, а потом выдает их со скоростью, соответствующей скорости второго процессора. И наоборот,

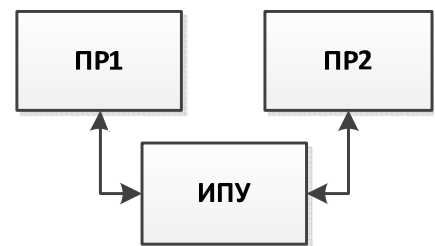


Рис. 9.6. Комплексирование «процессор – процессор»

принимая данные с меньшей скоростью, транслирует их быстрее. Данное комплексирование используется для одинаковых элементов системы.

Метод общей оперативной памяти

В вычислительной системе имеется общая память, к которой имеют доступ все процессоры системы (рис. 9.7). В случае необходи-

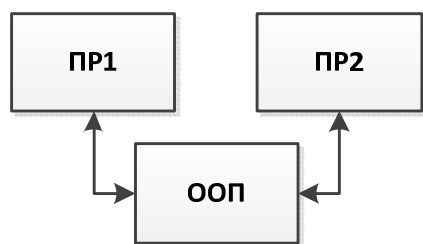


Рис. 9.7. Комплексирование по методу общей оперативной памяти

мости взаимодействия соответствующий процессор размещает информацию в этой памяти, а процессор – получатель информации – обращается к памяти и считывает предназначенную ему информацию. В идеале общая память должна размещаться в наиболее мощной машине, но теоретически может размещаться в любой. Вторым вариантом – создание специальной общей па-

мяти в виде отдельного устройства, и эта общая память специальными каналами связи соединена с процессорными элементами. В многомашинных системах память организуется любым образом, а в многопроцессорных, как правило, – отдельно.

При комплексировании через общую память нужно решить две задачи:

1. Организация взаимодействия между процессорами.
2. Разрешение конфликтов в том случае, если несколько процессоров попытаются обратиться к памяти.

Во втором случае проблема решается несколькими путями:

1. *Приоритетный принцип* – каждому процессору присваивается свой приоритет. Процессор с большим приоритетом имеет преимущество при доступе к памяти. Он может приостанавливать доступ к памяти процессорам с меньшим приоритетом. Этот принцип используется очень широко: в обычной машине процессор имеет наибольший приоритет.

2. *Распределение приоритетов*. В этом случае жесткое распределение приоритетов в вычислительных системах может оказаться некорректным, нецелесообразным, а динамическое распределение приоритетов (во времени) – сложным.

3. *Использование многовходовой памяти.* Существует несколько независимых друг от друга каналов ввода/вывода. Главной проблемой в данном случае является то, что многовходовая память дороже, её емкость и возможности ограничены, могут применяться при построении больших вычислительных систем.

При использовании общей памяти в качестве средства комплексирования наиболее часто используют память, организованную по *принципу почтового ящика*.

Комплексирование на уровне каналов ввода/вывода

На схеме (рис. 9.8) приведены: ПР1, ПР2 – процессоры, каналы в/в – каналы ввода/вывода; МК – мультиплексные каналы; СК – селективные каналы.

Мультиплексные каналы – это каналы с малой скоростью обмена, они обслуживают несколько источников или приемников. СК – каналы только для одного источника или только для одного приемника – это скоростной канал. Обмен между процессорами идет только че-

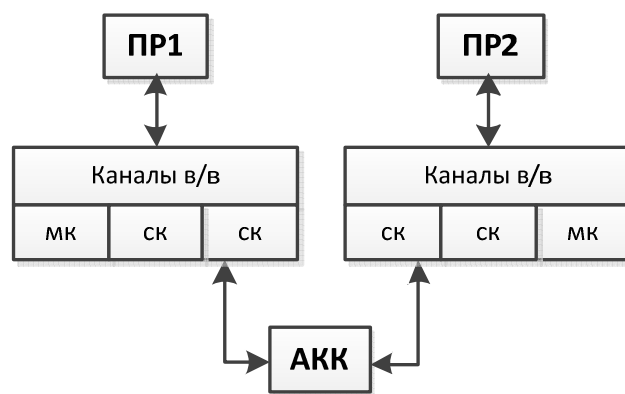


Рис. 9.8. Комплексирование на уровне каналов ввода/вывода

рез СК, а поскольку СК могут иметь различные характеристики, то между каналами ввода/вывода включается специальное устройство – адаптер канал – канал (АКК). АКК согласует характеристики каналов ввода/вывода отдельных процессов. Главная задача АКК – согласование скоростей обмена. В зависимости от соотношения скоростей он играет роль либо ускорителя, либо замедлителя. В любом случае внутри АКК есть буферное запоминающее устройство, с помощью которого происходит изменение скорости обмена. Если нужно замедлить скорость, то информация помещается в буфер и потом из него считывается в более медленном темпе.

Комплексирование на уровне управления внешними устройствами

Суть данного метода – любому процессору или любой машине системы обеспечить доступ к внешним устройствам другой машины или другого процессора. Каждый процессор через селекторный канал

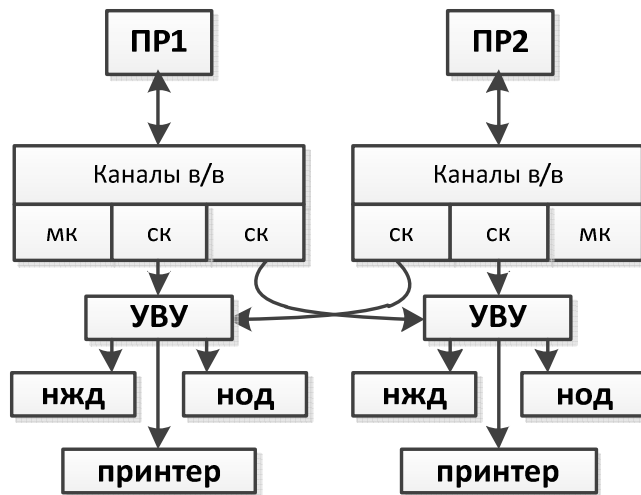


Рис. 9.9. Комплексирование на уровне управления внешними устройствами

получает доступ к устройству управления внешними устройствами и через это устройство доступ к соответствующим внешним устройствам (рис. 9.9). В этом случае либо расширяется номенклатура внешних устройств, либо можно выполнять какие-то процедуры ввода/вывода с параметрами, недоступными данному процессору. Данный способ комплексирования носит вспомогательный характер (расширяет периферию внешних устройств, но непосредственно на саму обработку данных этот способ комплексирования не влияет).

Комплексирование на уровне непосредственно внешних устройств

Существуют уникальные и достаточно дорогие внешние устройства, которые в вычислительной системе могут использоваться достаточно редко. Как правило, такие уникальные устройства низкоскоростные, поэтому для всех процессоров или машин в системе обеспечивается доступ к устройству не через

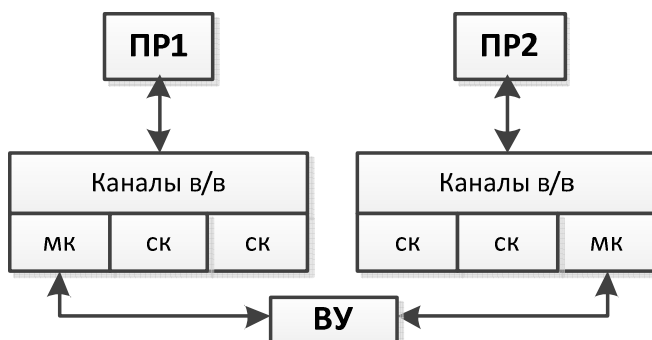


Рис. 9.10. Комплексирование на уровне непосредственно внешних устройств

скоростной селекторный канал, а через медленный мультиплексный канал (рис. 9.10). Схема со-

единения достаточно простая, так как вероятность конфликтов при одновременном обращении к этому устройству практически нулевая. И даже если конфликт произошел, то он разрешается просто организацией очереди (кто первый, тот первый и обслуживается, а остальные ждут). В данном случае очередь – не критический фактор.

Как правило, в реальных системах в чистом виде одно данное комплексирование не встречается – обязательно комбинация (вплоть до всех пяти методов комплексирования сразу).

9.6. Суперкомпьютеры и особенности их архитектуры

К суперкомпьютерам относятся мощные многопроцессорные вычислительные машины с быстродействием более десятка миллиардов операций в секунду.

Типовая модель *суперкомпьютера*:

- высокопараллельная многопроцессорная вычислительная система с быстродействием выше порядка 100 000 миллионов операций в секунду (*MFloPS*);
- емкость: оперативной памяти более 10 Гбайт, дисковой памяти более 10 Тбайт (1 Тбайт = 1000 Гбайт);
- разрядность 64 – 128 бит.

Создать такие высокопроизводительные компьютеры на одном МП не представляется возможным ввиду ограничения, обусловленного конечным значением скорости распространения электромагнитных волн (300 000 км/с), ибо время распространения сигнала на расстояние несколько миллиметров (линейный размер стороны МП) при быстродействии 100 млрд оп/с становится соизмеримым со временем выполнения одной операции. Поэтому суперкомпьютеры создаются в виде высокопараллельных многопроцессорных вычислительных систем (МПВС).

Высокопараллельные МПВС имеют несколько разновидностей.

1. Магистральные (конвейерные) МПВС, у которых процессор одновременно выполняет разные операции над последовательным потоком обрабатываемых данных. По принятой классификации такие МПВС относятся к системам с многократным потоком команд и однократным потоком данных (МКОД, или *MISD – Multiple Instruction Single Data*).

2. Векторные МПВС, у которых все процессоры одновременно выполняют одну команду над различными данными – однократный поток команд с многократным потоком данных (ОКМД, или *SIMD* – *Single Instruction Multiple Data*).

3. Матричные МПВС, у которых микропроцессор одновременно выполняет разные операции над последовательными потоками обрабатываемых данных – многократный поток команд с многократным потоком данных (МКМД, или *MIMD* – *Multiple Instruction Multiple Data*).

В суперкомпьютере используются все три варианта архитектуры МПВС:

- структура *MIMD* в классическом ее варианте (например, в суперкомпьютере *BSP* фирмы *Burroughs*);
- параллельно-конвейерная модификация, иначе *MMISD*, т.е. многопроцессорная (*Multiple*) *MISD*-архитектура (например, в суперкомпьютерах «Эльбрус 3, 4»);
- параллельно-векторная модификация, иначе *MSIMD*, т.е. многопроцессорная *SIMD*-архитектура (например, в суперкомпьютере *Cray 2*).

Наибольшую эффективность показала *MSIMD*-архитектура, поэтому в современных суперкомпьютерах чаще всего используется именно она (суперкомпьютеры фирм *Cray*, *Fujitsu*, *NEC*, *Hitachi* и т. д.). Первый суперкомпьютер был задуман в 1960 г. и создан в 1972 г. (машина *ILLIAC IV* с производительностью 20 *MFloPS*), а начиная с 1974 г. лидерство в разработке суперкомпьютеров захватила фирма *Cray Research*, выпустившая *Cray 1* производительностью 160 *MFloPS* и объемом оперативной памяти 64 Мбайта, а в 1984 г. – *Cray 2*, который в полной мере реализовал архитектуру *MSIMD* и ознаменовал появление нового поколения суперкомпьютеров. Производительность *Cray 2* – 2000 *MFloPS*, объем оперативной памяти – 2 Гбайта. Классическое соотношение, ибо критерий сбалансированности ресурсов компьютера – «каждому *MFloPS* производительности процессора должно соответствовать не менее 1 Мбайта оперативной памяти».

В настоящее время в мире насчитывается несколько тысяч суперкомпьютеров, начиная от простых офисных *Cray EL* до мощных *Cray 3*, *Cray 4*, *Cray Y-MPC90* фирмы *Cray Research*, *Cyber 205* фирмы *Control Data*, *SX-3* и *SX-X* фирмы *NEC*, *VP 2000* фирмы *Fujitsu* (обе фирмы японские), *VPP 500* фирмы *Siemens* (ФРГ) и так далее, производительностью несколько десятков тысяч *MFloPS*.

Среди лучших суперкомпьютеров можно отметить и отечественные. В сфере производства суперкомпьютеров Россия, пожалуй, впервые представила собственные оригинальные модели компьютеров (все остальные, включая и ПЭВМ, и малые ЭВМ, и универсальные компьютеры, за редким исключением, например, ЭВМ «Рута НО», копировали зарубежные разработки и в первую очередь разработки фирм США).

В СССР, а позднее в России была создана и реализуется государственная программа разработки суперкомпьютеров. В рамках этой программы были спроектированы и выпущены суперкомпьютеры, повторяющие архитектуру *Cray*: «Электроника СС БИС»; оригинальные разработки: ЕС 1191, ЕС 1195, ЕС 1191.01, ЕС 1191.10, «Эльбрус 1, 2, 3». Офисные варианты ЕС 1195, ЕС 1191.01 имеют производительность соответственно 50 *MFloPS* и 500 *MFloPS*.

Российской компанией «Т-Платформы» в 2009 г. был создан для МГУ им. М. В. Ломоносова суперкомпьютер «Ломоносов». В нём используются три вида вычислительных узлов и процессоры с различной архитектурой. В качестве основных узлов, обеспечивающих свыше 90 % производительности системы (а это более 420 триллионов операций в секунду (*TFloPS*)), используется *blade*-платформа *T-Blade2*. Суперкомпьютерные ресурсы МГУ используются в первую очередь для выполнения фундаментальных научных исследований, предполагающих ресурсоемкие вычисления. Среди таких задач – масштабные работы по глобальному изменению климата и динамике мирового океана, постгеномной медицине, механизмам формирования галактик и др.

Кластерные суперкомпьютеры

Как уже упоминалось выше, в настоящее время развивается технология построения больших и суперкомпьютеров на базе кластерных решений. По мнению многих специалистов, на смену отдельным, независимым суперкомпьютерам должны прийти группы высокопроизводительных серверов, объединяемых в кластер.

Удобство построения кластерных ВС заключается в том, что можно гибко регулировать необходимую производительность системы, подключая к кластеру с помощью специальных аппаратных и программных интерфейсов обычные серийные серверы до тех пор,

пока не будет получен суперкомпьютер требуемой мощности. Кластеризация позволяет манипулировать группой серверов как одной системой, упрощая управление и повышая надежность.

Важной особенностью кластеров является обеспечение доступа любого сервера к любому блоку как оперативной, так и дисковой памяти. Эта проблема успешно решается, например, объединением систем *SMP*-архитектуры на базе автономных серверов для организации общего поля оперативной памяти и использованием дисковых систем *RAID* для памяти внешней (*SMP* – *Shared Memory multiprocessing*, технология мультипроцессирования с разделением памяти).

Программное обеспечение для кластерных систем уже выпускается. Примером может служить компонент *Cluster Server* операционной системы *MS Windows NT/2000 Enterprise*. Этот компонент, более известный под кодовым названием *Wolfpack*, обеспечивает как функции управления кластером, так и функции диагностирования сбоев и восстановления (*Wolfpack* определяет сбой программы или отказ сервера и автоматически переключает поток вычислений на другие работоспособные серверы).

Несколько фирм (*Deli, SunMicrosystem, IBM*) уже продемонстрировали свои достижения в области суперкомпьютерных кластерных технологий (фирма *IBM*, например, продемонстрировала модель человеческого сердца, реализованную в кластере серверов *RS/6000*). Эта же фирма представила одну из самых мощных кластерных систем *ASCI White*, содержащую 8192 микропроцессора *IBM Power 3*, основную память емкостью 6 Тбайт, дисковую память емкостью 160 Тбайт; общая производительность кластера 12,5 *TFloPS*. Самый мощный в мире кластерный суперкомпьютер *K* расположен в Институте физико-химических исследований (*RIKEN*) в японском городе Кобе. Его современная конфигурация объединяет 864 вычислительные стойки и 705 тысяч процессорных ядер архитектуры *Sparc64*. Все фирмы отмечают существенное снижение стоимости кластерных систем по сравнению с локальными суперкомпьютерами, обеспечивающими ту же производительность. Многие исследователи считают, что использование кластерных систем – основной магистральный путь развития вычислительной техники новых поколений [11].

Вопросы к компьютерному тестированию

1. Какие свойства относятся к особенностям фон-неймановской архитектуры?
2. Какие тенденции являются причиной кризиса фон-неймановской архитектуры?
3. Каково назначение вычислительной системы?
4. Назовите основную отличительную особенность вычислительной системы от одиночной ЭВМ.
5. Каким соотношением связаны стоимость и производительность вычислительной системы?
6. Какие виды параллелизма заложены в классификации архитектур ВС по Флинну?
7. В какой из основных архитектур ВС существует необходимость пересылки «процессор – память»?
8. Какова основная особенность архитектуры *CISC*?
9. Какова основная особенность архитектуры *RISC*?
10. Какое время расходуется на обращение к памяти в архитектуре *BC RISC*, если время вычисления равно 1 мс?
11. В какой из основных архитектур ВС реализуется конвейерная организация вычислительного процесса?
12. Какую организацию должны иметь системы типа МКМД с массовым параллелизмом?
13. Что понимают под кластеризацией?
14. Запишите готовность системы в процентах при 10-минутном отказе в течение полуторачасовой работы.
15. В каком из вариантов многопроцессорных систем достигается наивысшая надежность вычислений?
16. Какой из уровней совместимости означает, что при правильном соединении информационных линий (кабелей) система должна сразу работать?
17. Что должно быть обеспечено при обмене данными в случае программной совместимости?

18. В каком из известных методов комплексирования процессоры по производительности должны быть согласованы?
19. В каком из известных методов комплексирования необходимо решать задачу взаимодействия между процессорами в отношении конфликтов?
20. Назовите методы, которыми разрешаются конфликты при одновременном обращении к общей оперативной памяти.
21. В чем преимущество процессора с большим приоритетом в случае приоритетного принципа разрешения конфликтов?
22. Что применяют при разрешении конфликтов с использованием многоходовой памяти?
23. Как называют каналы (полное название), по которым осуществляется комплексирование на уровне каналов ввода/вывода?
24. Какова главная задача адаптера «канал – канал» в случае комплексирования на уровне каналов ввода/вывода?
25. Какова главная цель комплексирования на уровне управления внешними устройствами?
26. Как называют каналы (полное название), по которым осуществляется комплексирование непосредственно на уровне внешних устройств?
27. Какое вспомогательное средство используют при возникновении конфликта в случае комплексирования непосредственно на уровне внешних устройств?
28. Как называют разновидность высокопараллельных многопроцессорных вычислительных систем, у которых процессор одновременно выполняет разные операции над последовательным потоком обрабатываемых данных?
29. Как называют разновидность высокопараллельных многопроцессорных вычислительных систем, у которых все процессоры одновременно выполняют одну команду над различными данными?
30. Как называют разновидность высокопараллельных многопроцессорных вычислительных систем, у которых микропроцессор одновременно выполняет разные операции над последовательными потоками обрабатываемых данных?

ЗАКЛЮЧЕНИЕ

Высокие темпы совершенствования вычислительной техники обусловлены ее важной ролью во всех сферах человеческой деятельности, в решении задач информатизации общества, обеспечении перехода от индустриального общества к информационному. Главная перспектива ее развития в настоящее время – дальнейшее расширение сфер применения ЭВМ и как следствие – переход от отдельных машин к вычислительным системам и комплексам разнообразных конфигураций с широким диапазоном функциональных возможностей и характеристик.

Так на какой же основе будет построена вычислительная система будущего? С достоверностью известно, что уже сейчас существуют системы обработки информации, построенные на объединении оптических и нейронных компьютеров, – это так называемые нейроно-оптические компьютеры. Для того чтобы создать мощную систему обработки информации, пришлось разработать гибридную систему, т.е. имеющую свойства как оптических, так и нейронных компьютеров. Можно предположить, что объединение квантовых и нейроно-оптических компьютеров даст миру самую мощную гибридную вычислительную систему. Такую систему от обычной будут отличать высокая производительность (за счет параллелизма) и возможность эффективной обработки и управления сенсорной информацией. Это, конечно, лишь предположение, которое никакими фактическими доказательствами в настоящее время не подкреплено, но технология создания вычислительных систем не стоит на месте, и в ближайшем будущем на рынке возможно появление новых вычислительных систем.

В последние годы при разработке и создании ЭВМ приоритет имеют сверхмощные компьютеры – суперЭВМ и сверхминиатюрные ПК.

Широкое внедрение средств мультимедиа (аудио- и видеосредств ввода и вывода информации) позволяет общаться с компьютером на естественном языке. Компьютеры стремительно обучаются мастерски узнавать лица, прослеживать взгляд и даже чувствовать настроение. Именно вокруг таких биометрических технологий построена исследо-

вательская программа *IBM* под названием *Blue Eyes* («Голубые глаза»). Еще один проект в рамках программы *Blue Eyes* – компьютерная мышь, способная чувствовать настроение: она будет оценивать эмоциональное состояние пользователя, измеряя через кончики пальцев его пульс, температуру и кожно-гальванический рефлекс. Возможные применения такого устройства относятся, по словам разработчиков, к областям видеоигр и изучения рынка.

Компания *Lexmark*, производящая принтеры, в настоящее время разрабатывает в сотрудничестве с Колледжем изящных искусств университета шт. Кентукки дизайн рабочего места для офиса будущего. Среди стильных компонентов – просвечивающий монитор, складная беспроводная клавиатура (похожая по виду на «высокотехнологичную» сумочку), которую можно носить по всему зданию и использовать с любым ПК, принтер, выпрыскивающий из картриджа и краску, и бумагу, а также «умный» стол, чувствующий приближение пользователя и подстраивающийся под его рост.

Специалисты предсказывают возможность создания компьютерной модели реального мира, такой виртуальной (кажущейся, воображаемой) системы, в которой мы можем активно жить и манипулировать виртуальными предметами.

Простейший прообраз такого кажущегося мира уже сейчас существует в сложных компьютерных играх. Но в будущем можно говорить не об играх, а о виртуальной реальности в нашей повседневной жизни, когда нас в комнате, например, будут окружать сотни активных компьютерных устройств, автоматически включающихся и выключающихся по мере надобности, отслеживающих наше местоположение, постоянно снабжающих нас необходимой информацией, воспринимающих нашу информацию и управляющих большинством окружающих приборов и устройств.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Айден, К. Аппаратные средства РС / К. Айден, Х. Фибельман, М. Крамер. – СПб. : ВHV-Санкт-Петербург, 1998. – 608 с. – ISBN 5-7791-0072-1.
2. Баула, В. Г. Введение в архитектуру ЭВМ и системы программирования : учеб. пособие / В. Г. Баула. – М. : Изд-во МГУ, 2003. – 144 с.
3. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации / В. Л. Бройдо, О. П. Ильина. – СПб. : Питер, 2011. – 560 с. – ISBN 978-5-49807-875-5.
4. Гук, М. Аппаратные средства IBM PC : энциклопедия / М. Гук. – СПб. : Питер, 2006. – 1072 с. – ISBN 978-5-469-01182-8.
5. Дорот, В. Л. Толковый словарь современной компьютерной лексики / В. Л. Дорот, Ф. А. Новиков. – 3-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2004. – 608 с. – ISBN 5-94157-491-6.
6. Информатика: Базовый курс / под ред. С. В. Симонович [и др.]. – СПб. : Питер, 2003. – 640 с. – ISBN 5-8046-0134-2.
7. Информатика : учебник / под ред. Н. В. Макаровой. – М. : Финансы и статистика, 2000. – 768 с. – ISBN 5-279-02202-0.
8. Колесниченко, О. В. Аппаратные средства РС / О. В. Колесниченко, И. В. Шишигин, В. Г. Соломенчук. – 6-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2010. – 682 с. – ISBN 978-5-9775-0432-4.
9. Корнеев, В. В. Параллельные вычислительные системы : монография / В. В. Корнеев. – М. : Нолидж, 1999. – 320 с. – ISBN 5-89251-065-4.
10. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – М. : Горячая линия-Телеком, 2002. – 382 с. – ISBN 5-93517-031-0.
11. Кручин, С. Суперкомпьютеры / С. Кручин // И&S. – 1995. – № 4.
12. Леонтьев, В. П. Новейшая энциклопедия компьютера 2011 / В. П. Леонтьев. – М. : ОЛМА Медиа Групп, 2011. – 960 с. – ISBN 978-5-373-03920-8.

13. Леонтьев, В. П. Персональный компьютер. Универсальный справочник пользователя 2001 / В. П. Леонтьев. – М. : ОЛМА-ПРЕСС, 2000. – 589 с. – ISBN 5-224-00404-7.
14. Мелехин, В. Ф. Вычислительные машины, системы и сети / учеб. для студентов высш. учеб. заведений / В. Ф. Мелехин, Е. Г. Павловский. – 3-е изд., перераб. и доп. – М. : Академия, 2010. – 560 с. – ISBN 978-5-7695-5840-5.
15. Нортон, П. Персональный компьютер: аппаратно-программная организация : пер. с англ. / П. Нортон, Дж. Гудман. – СПб. : ВНУ-Санкт-Петербург, 2000. – 848 с. – ISBN 5-7791-0111-6.
16. Основы современных компьютерных технологий : учеб. пособие / под ред. проф. А. Д. Хомоненко. – СПб. : КОРОНА-принт, 2009. – 448 с. – ISBN 978-5-7931-0494-4.
17. Полунов, Ю. Л. От абака до компьютера: судьбы людей и машин. Книга для чтения по истории вычислительной техники. В 2 т. Т. 1 / Ю. Л. Полунов. – М. : Русская редакция, 2004. – 480 с. – ISBN 5-7502-0078-7.
18. Попов, И. И. Электронные вычислительные машины и системы : учеб. пособие / И. И. Попов, Т. Л. Партыка. – М. : Инфра-М, 2014. – 368 с. – ISBN 978-5-91134-140-4.
19. Пятибратов, А. П. Вычислительные машины, сети и телекоммуникационные системы / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; под ред. проф. А. П. Пятибратова. – М. : ЕАОИ, 2009. – 292 с. – ISBN 978-5-374-00108-2.
20. Пятибратов, А. П. Вычислительные системы, сети и телекоммуникации / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко ; под ред. проф. А. П. Пятибратова. – 4-е изд., перераб. и доп. – М. : Инфра-М, 2014. – 736 с. – ISBN 978-5-279-03285-3.

ГЛОССАРИЙ

Адаптер

Устройство связи компьютера с периферийными устройствами.

Адрес

Номер конкретного байта оперативной памяти компьютера.

Алгебра логики

Раздел математики, изучающий высказывания, рассматриваемые со стороны их логических значений (истинности или ложности) и логических операций над ними.

Алгоритм

Заранее заданное понятное и точное предписание возможному исполнителю совершить определенную последовательность действий для получения решения задачи за конечное число шагов.

Алфавит

Фиксированный для данного языка набор основных символов, т.е. "букв алфавита", из которых должен состоять любой текст на этом языке. Никакие другие символы в тексте не допускаются.

Антивирусные программы

Программы, предотвращающие заражение компьютерным вирусом и ликвидирующие последствия заражения.

Арифметико-логическое устройство (АЛУ)

Часть процессора, которая производит выполнение операций, предусмотренных данным компьютером.

Архитектура фон Неймана

Архитектура компьютера, имеющего одно арифметико-логическое устройство, через которое проходит поток данных, и одно устройство управления, через которое проходит поток команд.
См. также: принципы фон Неймана.

Архитектура компьютера

Логическая организация, структура и ресурсы компьютера, которые может использовать программист. Определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера.

Ассемблер

См. Язык ассемблера.

ASCII

Читается "аски". Американский стандартный код обмена информацией. Широко используется для кодирования в виде байта букв, цифр, знаков операций и других компьютерных символов.

Аудиоадаптер (*Sound Blaster*, звуковая плата)

Специальная электронная плата, которая позволяет записывать звук, воспроизводить его и создавать программными средствами с помощью микрофона, наушников, динамиков, встроенного синтезатора и другого оборудования.

Байт

Группа из восьми битов, рассматриваемая при хранении данных как единое целое.

Библиотека стандартных подпрограмм

Совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.

Бит

Наименьшая единица информации в цифровом компьютере, принимающая значения «0» или «1».

Ввод

Считывание информации с внешнего устройства в память компьютера.

Вещественное число

В информатике – тип данных, содержащий числа, записанные с десятичной точкой и (или) с десятичным порядком.

Видеоадаптер

Электронная плата, которая обрабатывает видеоданные (текст и графику) и управляет работой дисплея. Содержит видеопамять, регистры ввода-вывода и модуль *BIOS*. Посылает в дисплей сигналы управления яркостью лучей и сигналы развёртки изображения.

Винчестер

См. Накопитель на жёстких магнитных дисках.

Внешняя память

Совокупность запоминающих устройств для длительного хранения данных. В состав внешней памяти входят накопители на гибких и жестких магнитных дисках, оптические и магнито-оптические накопители, накопители на магнитной ленте. Во внешней памяти обычно хранятся архивы программ и данных. Информация, размещенная на внешних носителях, не зависит от того, включен или выключен компьютер.

Вывод

Результаты работы программы, выдаваемые компьютером пользователю, другому компьютеру или во внешнюю память.

Выражение

В языке программирования – запись правила для вычисления некоторого значения. Строится из констант, переменных и указателей функций, объединенных знаками операций.

Гибкий (флоппи) диск

Круглая пластиковая пластина, покрытая с обеих сторон магнитным окислом и помещенная в защитную оболочку. Используется как носитель небольших объемов информации.

Графический редактор

Программа или комплекс программ, позволяющих создавать и редактировать изображения на экране компьютера: рисовать линии, раскрашивать области экрана, создавать надписи различными шрифтами, обрабатывать изображения, полученные с помощью сканеров. Некоторые редакторы обеспечивают возможность получения изображений трёхмерных объектов, их сечений и разворотов.

Двоичное кодирование

Способ представления данных в одном разряде в виде комбинации двух знаков, обычно обозначаемых цифрами «0» и «1». Разряд в этом случае называется двоичным разрядом.

Диск

Круглая металлическая или пластмассовая пластина, покрытая магнитным материалом, на которую информация наносится в виде концентрических дорожек, разделённых на секторы.

Дисковод

Устройство, управляющее вращением магнитного диска, чтением и записью данных на нём.

Дисплей

Устройство визуального отображения информации (в виде текста, таблицы, рисунка, чертежа и др.) на экране.

Дистрибутив

Форма распространения программного обеспечения, которая обычно содержит программы для начальной инициализации системы.

Домен

Область намагниченности в ферромагнитном кристалле или спонтанной поляризации в оптическом кристалле.

Драйверы

Программы, расширяющие возможности операционной системы по управлению устройствами ввода-вывода, оперативной памятью и т.д.; с помощью драйверов возможно подключение к компьютеру новых устройств или нестандартное использование имеющихся устройств.

Импликация

Импликация (лат. *implicatio* – связь) – бинарная логическая связка, по своему применению приближенная к союзам «если... то...».

Инструментальные программные средства

Программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.

Интегральная схема

Реализация электронной схемы, выполняющей некоторую функцию, в виде единого полупроводникового кристалла, в котором изготовлены все компоненты, необходимые для осуществления этой функции.

Интегрированные пакеты программ

Пакеты программ, выполняющие ряд функций, для которых ранее создавались специализированные программы, в частности, текстовые редакторы, электронные таблицы, системы управления базами данных, программы построения графиков и диаграмм.

Интерпретатор

Разновидность транслятора. Переводит и выполняет программу с языка высокого уровня в машинный код строка за строкой.

Интерфейс

Совокупность унифицированных технических и программных средств и правил (описаний, соглашений, протоколов), обеспечивающих одновременное взаимодействие устройств и/или программ в вычислительной системе или сопряжение между системами.

Каталог (директория, папка)

Оглавление файлов. Доступен пользователю через командный язык операционной системы. Его можно просматривать, переименовывать зарегистрированные в нём файлы, переносить их содержимое на новое место и удалять. Часто имеет иерархическую структуру.

Команда

Описание элементарной операции, которую должен выполнить компьютер. Обычно содержит код выполняемой операции, указания по определению операндов (или их адресов), указания по размещению получаемого результата. Последовательность команд образует программу.

Компакт-диск (CD-ROM)

Постоянное ЗУ, выполненное с использованием специальной оптической технологии. В ряду запоминающих устройств занимает место между флоппи- и жёстким дисками, являясь одновременно и мобильным, и очень ёмким.

Компилятор

Разновидность транслятора. Читает всю программу целиком, делает её перевод и создаёт законченный вариант программы на машинном языке, который затем и выполняется.

Контроллер

Устройство, которое связывает периферийное оборудование или каналы связи с центральным процессором, освобождая процессор от непосредственного управления функционированием данного оборудования.

Кэш

См. Сверхоперативная память.

Логический тип

Тип данных, представляемый значениями "истина" или "ложь" ("да" или "нет"). Иногда также называется булевым в честь английского математика XIX века Джорджа Буля.

Логический элемент (вентиль)

Часть электронной логической схемы, выполняющая элементарную логическую функцию.

Логическое высказывание

Любое предложение, в отношении которого можно однозначно сказать, истинно оно или ложно.

Магнитная головка

Устройство для записи, стирания и считывания информации с магнитного носителя: ленты или диска (жесткого или гибкого), представляющее собой катушку индуктивности, имеющую сердечник с магнитным зазором.

Массив

Последовательность однотипных элементов, число которых фиксировано и которым присвоено одно имя. Компьютерный эквивалент таблицы. Положение элемента в массиве однозначно определяется его индексами.

Машинный язык

Совокупность машинных команд компьютера, отличающаяся количеством адресов в команде, назначением информации, задаваемой в адресах, набором операций, которые может выполнить машина, и др.

Микропроцессор

Процессор, выполненный в виде интегральной схемы. Состоит из цепей управления, регистров, сумматоров, счётчиков команд, очень быстрой памяти малого объёма.

Накопитель на жёстких магнитных дисках (винчестерский накопитель)

Наиболее массовое запоминающее устройство большой ёмкости, в котором носителями информации являются круглые алюминиевые пластины – платтеры, обе поверхности которых покрыты слоем магнитного материала. Используется для постоянного хранения больших объёмов информации.

Операнд

Аргумент операции; данные, которые обрабатываются командой.

Оперативная память (ОЗУ)

Быстрое запоминающее устройство не очень большого объёма, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

Операционная система

Комплекс взаимосвязанных программ, предназначенных для автоматизации планирования и организации процесса обработки программ, ввода-вывода и управления данными, распределения ресурсов, подготовки и отладки программ, других вспомогательных операций обслуживания. Важнейшая часть программного обеспечения.

Основание системы счисления

Количество различных цифр, используемых для изображения чисел в данной системе счисления.

Отладчик (англ. *debugger*)

Программа, позволяющая исследовать внутреннее поведение разрабатываемой программы. Обеспечивает пошаговое исполнение программы с остановкой после каждого оператора, просмотр текущего значения переменной, нахождение значения любого выражения и др.

Пакеты прикладных программ (ППП)

Специальным образом организованные программные комплексы, рассчитанные на общее применение в определённой проблемной области и дополненные соответствующей технической документацией.

Параллельный код

Способ передачи цифровой информации, когда биты передаются поразрядно без временного разделения.

Персональный компьютер

Микрокомпьютер универсального назначения, рассчитанный на одного пользователя и управляемый одним человеком.

Перфокарта

Перфорационная карта, перфорированная карта (лат. *perforo* – пробиваю и лат. *charta* – лист из папируса); бумага – носитель информации, предназначенный для использования в системах автоматической обработки данных, изготовленная из тонкого картона, перфокарта представляет информацию наличием или отсутствием отверстий в определённых позициях карты.

Перфолента

Носитель информации в виде узкой тонкой ленты из бумаги или пластмассы, на которую информация записывается пробивкой отверстий (перфораций).

Подпрограмма

Самостоятельная часть программы, которая создаётся независимо от других частей и затем вызывается по имени. Когда имя подпрограммы используется в качестве оператора программы, выполняется вся группа операторов, представляющая тело подпрограммы.

Порты устройств

Электронные схемы, содержащие один или несколько регистров ввода-вывода и позволяющие подключать периферийные устройства компьютера к внешним шинам микропроцессора. Последовательный порт обменивается данными с процессором побайтно, а с внешними устройствами – побитно. Параллельный порт получает и посылает данные побайтно.

Последовательный код

Способ передачи цифровой информации, когда биты передаются с временным разделением и использованием синхронизации.

Постоянная память (ПЗУ)

Энергонезависимое запоминающее устройство, изготовленное в виде микросхемы. Используется для хранения данных, не требующих изменения. Содержание памяти специальным образом "зашивается" в ПЗУ при изготовлении. В ПЗУ находятся программа управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

Прикладная программа

Любая конкретная программа, способствующая решению какой-либо задачи в пределах данной проблемной области.

Принцип открытой архитектуры

1. Регламентируются и стандартизируются только описание принципа действия компьютера и его конфигурация (определённая совокупность аппаратных средств и соединений между ними). Таким образом, компьютер можно собирать из отдельных узлов и деталей, разработанных и изготовленных независимыми фирмами-изготовителями.
2. Компьютер легко расширяется и модернизируется за счёт наличия внутренних расширительных гнезд, в которые пользователь может вставлять разнообразные устройства, удовлетворяющие заданному стандарту, и тем самым устанавливать конфигурацию своей машины в соответствии с личными предпочтениями.

Принципы фон Неймана

1. *Принцип программного управления.* Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определённой последовательности.
2. *Принцип адресности.* Основная память состоит из перенумерованных ячеек; процессору времени доступна любая ячейка.
3. *Принцип однородности памяти.* Программы и данные хранятся в одной и той же памяти, поэтому компьютер не различает, что хранится в данной ячейке памяти – число, текст или команда. Над командами можно выполнять такие же действия, как и над данными.

Программное обеспечение (SoftWare)

Совокупность программ, выполняемых компьютером, а также вся область деятельности по проектированию и разработке программ.

Регистр

Специальная запоминающая ячейка с функциями кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличается от ячейки памяти тем, что может не только хранить двоичный код, но и преобразовывать его.

Регистр команд

Регистр устройства управления для хранения кода команды на период времени, необходимый для её выполнения.

Резонанс

Явление резкого возрастания амплитуды вынужденных колебаний, которое наступает при совпадении частоты собственных колебаний с частотой колебаний вынуждающей силы.

Сверхоперативная память

Очень быстрое ЗУ малого объёма. Используется для компенсации разницы в скорости обработки информации процессором и несколько менее быстродействующей оперативной памятью.

Сервер

Высокопроизводительный компьютер с большим объёмом внешней памяти, который обеспечивает обслуживание других компьютеров путем управления распределением дорогостоящих ресурсов совместного пользования (программ, данных и периферийного оборудования).

См. также: клиент.

Система команд

Совокупность операций, выполняемых некоторым компьютером.

Система программирования

Система для разработки новых программ на конкретном языке программирования. Предоставляет пользователю мощные и удобные средства разработки программ: транслятор, редактор текстов программ, библиотеки стандартных программ, отладчик и др.

Система счисления

Совокупность приемов и правил, по которым записываются и читаются числа.

Система управления базами данных (СУБД)

Система программного обеспечения, позволяющая обрабатывать обращения к базе данных, поступающие от прикладных программ конечных пользователей.

Системные программы

Программы общего пользования, выполняемые вместе с прикладными и служащие для управления ресурсами компьютера – центральным процессором, памятью, вводом-выводом.

Системы автоматизированного проектирования (САПР)

Комплексные программно-технические системы, предназначенные для выполнения проектных работ с применением математических методов. Широко используются в архитектуре, электронике, механике и др. В качестве входной информации в САПР используются технические знания специалистов, которые вводят проектные требования, уточняют результаты, проверяют полученную конструкцию, изменяют ее и т.д. В САПР накапливается информация, поступающая из библиотек стандартов (данные о типовых элементах конструкций, их размерах, стоимости и др.).

Стример

Устройство для резервного копирования больших объёмов информации. В качестве носителя применяются кассеты с магнитной лентой ёмкостью 1 – 2 Гбайта и больше.

Сумматор

Электронная логическая схема, выполняющая суммирование двоичных чисел.

Суперкомпьютер

Очень мощный компьютер с производительностью свыше 100 мегафлопов (1 мегафлоп – миллион операций с плавающей точкой в секунду). Представляет собой многопроцессорный и (или) многомашинный комплекс, работающий на общую память и общее поле внешних устройств. Архитектура основана на идеях параллелизма и конвейеризации вычислений.

Схема алгоритма (блок-схема)

Графическое представление алгоритма в виде последовательности блоков, соединённых стрелками.

Счётчик команд

Регистр УУ, содержимое которого соответствует адресу очередной выполняемой команды; служит для автоматической выборки команд программы из последовательных ячеек памяти.

Таблица истинности

Табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

Текстовый редактор

Программа для ввода и изменения текстовых данных (документов, книг, программ, ...). Обеспечивает редактирование строк текста, контекстный поиск и замену частей текста, автоматическую нумерацию страниц, обработку и нумерацию сносок, выравнивание краёв абзаца, проверку правописания слов и подбор синонимов, построение оглавлений, распечатку текста на принтере и др.

Тип данных

Понятие языка программирования, определяющее структуру констант, переменных и других элементов данных, разрешенные их значения и операции, которые можно над ними выполнять.

Транслятор

Программа-переводчик. Преобразует программу, написанную на одном из языков высокого уровня, в программу, состоящую из машинных команд.

Триггер

Электронная схема, широко применяемая в регистрах компьютера для надёжного запоминания одного бита информации. Имеет два устойчивых состояния, которые соответствуют двоичной «1» и двоичному «0».

Устройство управления (УУ)

Часть процессора, выполняющая функции управления устройствами компьютера.

Файл

Именованная совокупность любых данных, размещенная на внешнем запоминающем устройстве и хранимая, пересылаемая и обрабатываемая как единое целое. Файл может содержать программу, числовые данные, текст, закодированное изображение и др. Имя файла регистрируется в каталоге.

Формирователь

Устройство или программа, обеспечивающие создание сигнала заданной структуры.

Фоторезист

Полимерный светочувствительный материал. Наносится на обрабатываемый материал в процессе фотолитографии или фотогравировки с целью получить соответствующее фотошаблону расположение окон для доступа травящих или иных веществ к поверхности обрабатываемого материала.

Электронная таблица

Программа, обрабатывающая таблицы, состоящие из строк и граф, на пересечении которых располагаются клетки. В клетках содержатся числовая информация, формулы или текст. Значение в числовой клетке таблицы либо записано, либо рассчитано по формуле. В формуле могут присутствовать обращения к другим клеткам.

Юникод

Стандарт кодирования символов, предложенный в 1991 году некоммерческой организацией «Консорциум Юникода» и позволяющий представить знаки почти всех письменных языков.

Язык ассемблера

Система обозначений, используемая для представления в удобочитаемой форме программ, записанных в машинном коде. Перевод программы с языка ассемблера на машинный язык осуществляется специальной программой, которая называется *ассемблером* и является, по сути, простейшим транслятором.

Язык высокого уровня

Язык программирования, более близкий к естественному языку, чем машинный код или язык ассемблера. Каждый оператор в нём соответствует нескольким командам машинного кода или языка ассемблера.

Учебное издание

ГАЛАС Валерий Петрович

**ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ
И ТЕЛЕКОММУНИКАЦИИ**

Учебник

Часть 1. Вычислительные системы

Редактор А. А. Амирсейидова
Технический редактор С. Ш. Абдуллаева
Корректор Е. С. Глазкова
Компьютерная верстка Е. А. Герасиной
Дизайн обложки В. П. Галаса

Подписано в печать 22.12.16.

Формат 60×84/16. Усл. печ. л. 13,49. Тираж 60 экз.

Заказ

Издательство

Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых.
600000, Владимир, ул. Горького, 87.