ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального образования
Ульяновский государственный технический университет

Н. Г. Захаров, Р. А. Сайфутдинов

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Допущено учебно-методическим объединением высших военно-учебных заведений по образованию в области военного управления Вооруженных Сил Российской Федерации (Сухопутные войска) в качестве учебника для курсантов высших военно-учебных заведений, обучающихся по специальностям направлений «Телекоммуникации» и «Информатика и вычислительная техника»

Ульяновск
2007

УДК 32.973 (075)
ББК 681.324я7
3-38

Рецензент: доктор технических наук, заведующий кафедрой «Телекоммуникации» Ульяновского государственного технического университета, профессор Васильев К. К,; доктор технических наук, директор Ульяновского филиала института радиотехники и электроники РАН, профессор Сергеев В. А.

Захаров Н. Г.

З-38 Вычислительная техника: учебник / Н. Г. Захаров, Р, А. Сайфутдинов. - Ульяновск: УлГТУ, 2007. - 224 с.

ISBN 978-5-9795-0128-4

Настоящий учебник освещает арифметические и логические основы вычислительной техники, комбинационные и последовательные цифровые устройства, цифро-аналоговые и аналого-цифровые преобразователи. Рассматриваются принципы функционирования компьютеров, архитектура ПЭВМ, внешние запоминающие устройства н диагностика компьютеров.

Учебник предназначен для студентов высших учебных заведений н курсантов военных учебных заведений МО РФ, изучающих дисциплины «Вычислительная техника и информационные технологии», «Цифровые устройства и микропроцессоры» и «Схемотехника ЭВМ».

УДК 32.973(075)
ББК 681324я7

© Н. Г. Захаров, Р. А. Сайфутдинов, 2007
© Оформление. УлГТУ, 2007

ISBN 978-5-9795-0128-4

Оглавление

Введение **6**

Часть I. Цифровые устройства **8**

1. Арифметические основы цифровых устройств **8**

1.1. Системы счисления 8

1.2. Двоичная, восьмеричная и шестнадцатеричная системы счисления 11

1.3. Перевод числа из одной системы счисления в другую 15

Контрольные вопросы 17

2. Логические основы цифровых устройств **18**

2.1. Основы алгебры логики 18

2.2. Основные законы алгебры логики 18

2.3. Преобразование булевых выражений 20

2.4. Дизъюнктивные нормальные формы 20

2.5. Минимизация логических функций 22

2.6. Табличные методы минимизации. Карты Карно 23

2.7. Неполностью определенные логические функции 28

2.8. Логические элементы и логические операции 30

2.9. Классификация логических элементов 33

2.10. Основные характеристики логических элементов 34

2.11. Резисторно-транзисторные логические элементы 36

2.12. Транзисторные элементы с непосредственной связью (НСТЛ) 38

2.13. Транзисторно-транзисторные логические элементы 39

2.14. Логические элементы с тремя состояниями 44

2.15. ТТЛ логические элементы с открытым коллектором 45

2.16. ЭСЛ логические элементы 45

2.17. Логические элементы на МОП-транзисторах 48

Контрольные вопросы 52

3. Цифровые комбинационные устройства **53**

3.1. Устройство равнозначности 53

3.2. Устройство неравнозначности 55

3.3. Комбинационные сумматоры 56

3.4. Дешифраторы 58

3.5. Шифраторы 62
3.6. Мультиплексоры 65
3.7. Демультиплексоры 67
3.8. Преобразователи кодов 68
Контрольные вопросы 73
4. Последовательностные цифровые устройства 74
4.1. Триггеры. Общее определение 74
4.2. Классификация триггеров 75
4.3. Асинхронный RS-триггер с прямыми входами 77
4.4. Асинхронный RS-триггер с инверсными входами 78
4.5. Синхронный RS-триггер 79
4.6. D-триггер 82
4.7. Универсальный JK-триггер 84
4.8. T-триггер 86
4.9. Синхронный триггер с динамическим управлением 87
4.10. Функции возбуждения триггеров 88
4.11. Регистры 90
4.11.1. Регистры памяти 91
4.11.2. Сдвигающие регистры 92
4.12. Счетчики 95
4.12.1. Параллельные счетчики 97
4.12.2. Реверсивные счетчики 98
Контрольные вопросы 99
5. Цифро-аналоговые и аналогово-цифровые преобразователи 100
5.1. Цифро-аналоговые преобразователи 100
5.2. Аналого-цифровые преобразователи 104
Контрольные вопросы 110
Часть II. Принципы построения компьютеров 111
6. Общие принципы построения компьютеров 111
6.1. Этапы развития ЭВМ 111
6.2. Принципы фон Неймана 117
6.3. Принципы построения компьютеров 119

6.4. Центральный процессор 123
6.5. Формы представления чисел в ЭВМ 126
Контрольные вопросы 133
7. МикроЭВМ 134
7.1. Структурная схема микроЭВМ 134
7.2. Система команд и способы адресации 140
7.3. Типы команд микроЭВМ 143
7.4. Выполнение команды микропроцессором 145
7.5. Организация интерфейса в микроЭВМ 149
Контрольные вопросы 151
8. Архитектура ПЭВМ 152
8.1. Системный блок ПЭВМ 152
8.2. Основные характеристики микропроцессоров, используемых в ПЭВМ 157
8.3. Память ПЭВМ 164
8.4. Внешние устройства 171
8.5. Системные и периферийные шины ПЭВМ 179
Контрольные вопросы 189
9. Внешние запоминающие устройства 190
9.1. Классификация и характеристики внешних запоминающих устройств 190
9.2. Накопители на гибких магнитных дисках 191
9.3. Накопители на жестких магнитных дисках 195
9.4. Накопители на оптических дисках 197
9.5. Структура файловой системы Windows 202
Контрольные вопросы 206
10. Диагностика ПЭВМ 207
10.1. Программа POST 207
10.2. Главное меню программы BIOS 209
10.3. Тестовые программные средства 210
Контрольные вопросы 220
Список сокращений 221
Библиографический список 223

ВВЕДЕНИЕ

Вычислительная техника является основой построения информационно-измерительных систем, используемых для решения важнейших научно-технических задач. Вычислительные устройства обеспечивают моделирование реальных радиотехнических комплексов в различных ситуациях, работу систем автоматизированного проектирования, управление сложнейшими технологическими процессами. Проблемы вычислительной техники следующие: это создание и использование современной элементной базы, разработка цифровых узлов и устройств, входящих в состав компьютерных систем, а также разработка новых модифицированных компьютеров с нейронной структурой построения.

Знание элементной базы и эффективное использование ее для построения цифровых устройств в компьютерах обеспечивает успешную эксплуатацию и обслуживание средств вычислительной техники.

Учебник состоит из двух частей и 10 глав. Первая часть посвящена цифровым устройствам, вторая – принципам построения и функционирования компьютеров.

В первой главе рассмотрены арифметические основы цифровых устройств, системы счисления, перевод чисел из одной системы счисления в другую, выполнение арифметических операций в цифровых вычислительных устройствах.

Во второй главе изложены логические основы цифровых устройств. Рассмотрены основные законы алгебры логики, минимизация логических функций, логические операции и логические элементы, ТТЛ, ТТЛШ, ЭСЛ, КМОП логические элементы и их характеристики.

В третьей главе рассматриваются комбинационные устройства: сумматоры, шифраторы, дешифраторы, цифровые коммутаторы, преобразователи кодов.

Четвертая глава посвящена изучению последовательностных цифровых устройств. Рассмотрены триггеры RS, D, JK типов и счетчики на их основе. Приведены схемы статических и динамических триггеров, даны функции их возбуждения. Приведены схемы регистров памяти и сдвига.

Пятая глава посвящена цифро-аналоговым и аналого-цифровым преобразователям, приведены схемы, методы построения и условно-графические схемы преобразователей.

В шестой главе изложены общие принципы построения компьютеров, приведены этапы развития ЭВМ по поколениям развития и элементной базе, принципы фон Неймана, структура центрального процессора и формы представления целых и вещественных чисел в ЭВМ.

В седьмой главе рассмотрена структура микроЭВМ, система команд и способы адресации. Дается описание типов команд в микроЭВМ и пошаговое выполнение команды микропроцессором, а также организация интерфейсов.

В восьмой главе изложена функциональная и структурная организации ПЭВМ. Рассмотрен состав системного блока, основные характеристики микропроцессора, организация и характеристики памяти ПЭВМ, системный и периферийный интерфейсы, а также принцип работы устройств ввода и вывода информации.

В девятой главе даны классификация и характеристика внешних запоминающих устройств, структура накопителей на гибких и жестких магнитных дисках. Приведены классификация и принципы записи информации на оптических дисках, а также структура файловой системы Windows.

В десятой главе рассматриваются программа начальной загрузки компьютера POST, меню базовой системы ввода-вывода BIOS. Даются описание, состав и возможности тестовых программных средств.

Предлагаемый учебник позволяет приобрести теоретические знания и практические навыки по дисциплинам «Схемотехника ЭВМ», «Вычислительная техника и информационные технологии» и «Цифровые устройства и микропроцессоры» и базируется на изданиях, рекомендованных Министерством образования Российской Федерации для высших учебных заведений.

ЧАСТЬ 1. ЦИФРОВЫЕ УСТРОЙСТВА

1. АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ УСТРОЙСТВ

1.1. Системы счисления

Система счисления — совокупность приемов и правил наименования и обозначения чисел, позволяющих установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов.

В любой системе счисления выбирается алфавит, представляющий собой совокупность некоторых символов (букв или цифр), с помощью которого в результате каких-либо операций можно представить любое их количество. Изображение любого количества символов называется числом, а символы алфавита — буквами и цифрами. Символы алфавита должны быть разными и значение каждого из них должно быть известно.

В современном мире наиболее распространенной является десятичная система счисления, происхождение которой связано с пальцевым счетом. Она возникла в Индии и в XIII в. была перенесена в Европу арабами. Поэтому десятичную систему счисления стали называть арабской, а используемые для записи чисел цифры, которыми мы теперь пользуемся, — 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 — арабскими.

С давних времен для подсчетов и вычислений применялись различные системы счисления. Например, на Древнем Востоке довольно широко была распространена двенадцатеричная система. Многие предметы (ножи, вилки, тарелки и т. д.) и сейчас считают дюжинами. Число месяцев в году — двенадцать. Эта система счисления сохранилась в английской системе мер (например, 1 фут = 12 дюймов) и в денежной системе (1 шиллинг =12 пенсов). В Древнем Вавилоне существовала весьма сложная 60-ричная система. Она, как и 12-ричная система, в какой-то степени сохранилась и до наших дней (например, в системе измерения времени: 1 ч = 60 мин, 1 мин = 60 с, аналогично в системе измерения углов: 1° = 60′, 1′ = 60″).

Первые цифры (знаки для обозначения чисел) появились у египтян и вавилонцев. У ряда народов (древние греки, сирийцы, финикийцы) цифрами служили буквы

алфавита. Аналогичная система до XVI в. применялась и в России. В Средние века в Европе пользовались системой римских цифр, которые и сейчас часто применяют для обозначения глав, частей, разделов в различного рода документах, книгах, для обозначения месяцев и т. д.

Все системы счисления можно разделить на позиционные и непозиционные.

Непозиционная система счисления — система, в которой символы, обозначающие то или иное количество, не меняют своего значения в зависимости от местоположения (позиции) в изображении числа.

Запись числа A в непозиционной системе счисления D может быть представлена выражением

$$A_D = D_1 + D_2 + \dots D_N = \sum_{i=1}^{N} D_i ,$$

где A_D — запись числа A в системе счисления D;

D_i — символы системы.

Непозиционной системой счисления является самая простая система с одним символом (палочкой). Для изображения какого-либо числа в этой системе надо записать количество палочек, равное данному числу. Например, запись числа 12 в такой системе счисления будет иметь вид: IIIIIIIIIII. Эта система неэффективна, так как форма записи очень громоздка.

К непозиционной системе счисления относится и римская, символы алфавита которой и обозначаемое ими количество представлены в табл. 1.1.

Таблица 1.1

Римские цифры	I	V	X	L	C	D	M
Значение (обозначаемое количество)	1	5	10	50	100	500	1000

Запись чисел в этой системе осуществляется по следующим правилам:

1) если цифра слева меньше, чем справа, то левая цифра вычитается из правой (IV: 1 < 5, следовательно, 5 – 1 = 4, XL: 10 < 50, следовательно, 50 – 10 = 40);

2) если цифра справа меньше или равна цифре слева, то эта цифры складываются (VI: 5 + 1 = 6, VIII: 5 + 1 + 1 + 1 = 8, XX: 10 + 10 = 20).

Так, число 1964 в римской системе счисления имеет вид MCMLXIV (M – 1000, CM – 900, LX – 60, IV – 4), здесь «девятьсот» получается посредством вычитания из «тысячи» числа «сто», «шестьдесят» — посредством сложения «пятидесяти» и «десяти», «четыре» — посредством вычитания из «пяти» «единицы».

В общем случае непозиционные системы счисления характеризуются сложными способами записи чисел и правилами выполнения арифметических операций. В настоящее время все наиболее распространенные системы счисления относятся к разряду позиционных.

Позиционные системы счисления

Систему счисления, в которой значение цифры определяется ее местоположением (позицией) в изображении числа, называют ***позиционной.***

Упорядоченный набор символов (букв и цифр) $\{a_0, a_1, \ldots, a_n\}$, используемый для представления любых чисел в заданной позиционной системе счисления, называют ее алфавитом, число символов (цифр) алфавита $p = n + 1$ — ее основанием, а саму систему счисления называют p-ричной.

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

Самой привычной для нас является десятичная система счисления. Ее алфавит — {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, а основание $p = 10$, т. е. в этой системе для записи любых чисел используется только десять разных символов (цифр). Эти цифры введены для обозначения первых десяти последовательных чисел, а все последующие числа, начиная с 10 и т. д., обозначаются уже без использования новых цифр. Десятичная система счисления основана на том, что 10 единиц каждого разряда объединяются в одну единицу соседнего старшего разряда, поэтому каждый разряд имеет вес, равный степени 10. Следовательно, значение одной и той же цифры определяется ее местоположением в изображении числа, характеризуемым степенью числа 10. Например, в изображении числа 222,22 цифра 2 повторяется 5 раз, при этом первая слева цифра 2 означает количество сотен (ее вес равен 10^2); вторая — количество десятков (ее вес равен 10), третья — количество единиц (ее вес равен 10^0), четвертая — количество десятых долей единицы (ее вес равен 10^{-1}) и пятая цифра — количество сотых долей единицы (ее вес равен 10^{-2}), т. е. число 222,22 может быть разложено по степеням числа 10:

$\mathbf{222,22} = \mathbf{2} \cdot 10^2 + \mathbf{2} \cdot 10^1 + \mathbf{2} \cdot 10^0 + \mathbf{2} \cdot 10^{-1} + \mathbf{2} \cdot 10^{-2}$.

Аналогично

$\mathbf{725} = \mathbf{7} \cdot 10^2 + \mathbf{2} \cdot 10^1 + \mathbf{5} \cdot 10^0$;

$\mathbf{1304,5} = \mathbf{1} \cdot 10^3 + \mathbf{3} \cdot 10^2 + \mathbf{0} \cdot 10^1 + \mathbf{4} \cdot 10^0 + \mathbf{5} \cdot 10^{-1}$;

$\mathbf{50328,15} = \mathbf{5} \cdot 10^4 + \mathbf{0} \cdot 10^3 + \mathbf{3} \cdot 10^2 + \mathbf{2} \cdot 10^1 + \mathbf{8} \cdot 10^0 + \mathbf{1} \cdot 10^{-1} + \mathbf{5} \cdot 10^{-2}$.

Таким образом, любое число А можно представить в виде полинома путем разложения его по степеням числа 10:

$$A_{10} = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + ... + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + ... + a_{-m} \cdot 10^{-m} + ...,$$

последовательность из коэффициентов которого представляет собой десятичную запись числа A_{10}:

$$A_{10} = a_n\, a_{n-1} \ldots a_1\, a_0\, ,\, a_{-1} \ldots a_{-m} \ldots$$

Запятая, отделяющая целую часть числа от дробной, служит для фиксации конкретных значений каждой позиции в этой последовательности цифр и является началом отсчета.

1.2. Двоичная, восьмеричная и шестнадцатеричная системы счисления

Примеры изображения чисел в данных системах счисления представлены в табл. 1.2.

В современной вычислительной технике, в устройствах автоматики и связи используется в основном двоичная система счисления, что обусловлено рядом преимуществ данной системы счисления перед другими системами. Так, для ее реализации нужны технические устройства лишь с двумя устойчивыми состояниями, например материал намагничен или размагничен (магнитные ленты, диски), отверстие есть или отсутствует (перфолента и перфокарта). Этот метод обеспечивает более надежное и помехоустойчивое представление информации, дает возможность применения аппарата булевой алгебры для выполнения логических преобразований информации. Кроме того, арифметические операции в двоичной системе счисления выполняются наиболее просто.

Таблица 1.2

Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Недостаток двоичной системы — быстрый рост числа разрядов, необходимых для записи больших чисел. Этот недостаток не имеет существенного значения для ЭВМ. Если же возникает необходимость кодировать информацию, «вручную», например при составлении программы на машинном языке, то используют восьмеричную или шестнадцатеричную системы счисления. Числа в этих системах читаются почти так же легко, как десятичные, требуют соответственно в 3 (восьмеричная) и в 4 (шестнадцатеричная) раза меньше разрядов, чем в двоичной системе (числа 8 и 16 — соответственно 3-я и 4-я степени числа 2), а перевод их в двоичную систему счисления и обратно осуществляется гораздо проще в сравнении с десятичной системой счисления.

Арифметические операции в двоичной системе счисления

Правила выполнения арифметических действий над двоичными числами задаются таблицей сложения, вычитания и умножения (табл. 1.3).

Таблица 1.3

Сложение	Вычитание	Умножение
0 + 0 = 0	0 – 0 = 0	0 · 0 = 0
0 + 1 = 1	1 – 0 = 1	0 · 1 = 0
1 + 0 = 1	1 – 1 = 0	1 · 0 = 0
1 + 1 = 10	10 – 1 = 1	1 · 1 = 1

Единица – перенос в старший разряд

Правила арифметики во всех позиционных системах счисления аналогичны. В двоичной системе счисления арифметическое сложение происходит по правилу сложения по модулю два с учетом переноса единицы в старший разряд.

Пример. Выполнить операцию арифметического сложения в двоичной системе счисления чисел 13 и 7.

$13_{10} = 1101_2$ $\quad 7_{10} = 0111_2$

Решение:

$$\begin{array}{r} 13 \\ +\ 7 \\ \hline 20_{10} \end{array} \qquad \begin{array}{r} 01101 \\ +\ 00111 \\ \hline 10100_2 \end{array}$$

При сложении двух единиц результат операции равен нулю и единица переносится в соседний разряд.

$$\overset{4\ 3\ 2\ 1\ 0}{1\ 0\ 1\ 0\ 0}_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 20_{10}.$$

Пример. Выполнить операцию арифметического вычитания в двоичной системе счисления чисел 12 и 7.

Решение:

$$\begin{array}{r} 12 \\ -\ 7 \\ \hline 5_{10} \end{array} \qquad \begin{array}{r} \overset{1\ 11}{} \\ 1\ 1\ 0\ 0 \\ -\ 0\ 1\ 1\ 1 \\ \hline 0\ 1\ 0\ 1_2 \end{array}$$

При вычитании из нулевого разряда в данном разряде образуются две единицы, а в соседних нулевых разрядах возникает единица.

$$\overset{3\ 2\ 1\ 0}{0\ 1\ 0\ 1}_2 = 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5_{10}.$$

Таблицы сложения для восьмеричной и шестнадцатеричной систем счисления представлены на рис. 1.1 и 1.2.

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Рис. 1.1. Таблица сложения для восьмеричной систем счисления

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	**0**	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	**2**	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	**4**	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	**6**	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	**8**	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	**A**	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	**C**	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	**E**	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	**10**	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	**12**	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	**14**	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	**16**	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	**18**	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	**1A**	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	**1C**	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	**1E**

Рис. 1.2. Таблица сложения для шестнадцатеричной системы счисления

При сложении цифры суммируются по разрядам, и если при этом возникает избыток, то он переносится влево.

1.3. Перевод числа из одной системы счисления в другую

При переводе целого *десятичного* числа в систему с основанием *q* его необходимо последовательно *делить* на *q* до тех пор, пока не останется остаток, меньший или равный *q–1*. Число в системе с основанием *q* записывается как последовательность остатков от деления, записанных в обратном порядке, начиная с последнего.

Пример. Перевести число 75 из десятичной системы в двоичную, восьмеричную и шестнадцатеричную:

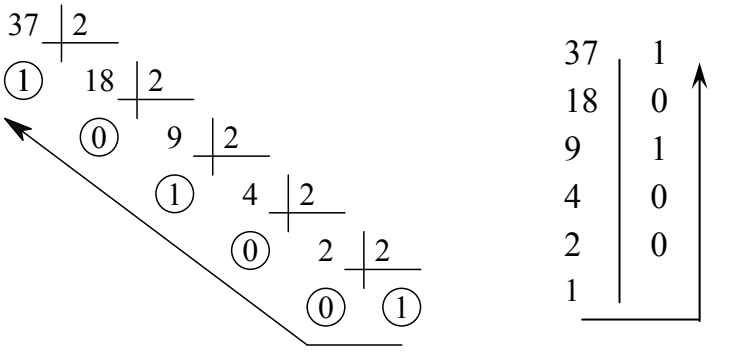
в двоичную *в восьмеричную* *в шестнадцатеричную*

Ответ: $75_{10} = 1\ 001\ 011_2 = 113_8 = 4B_{16}$.

Другой способ записи перевода числа из десятичной системы счисления в двоичную осуществляется следующим способом.

Исходное число делят на 2, результат пишут под исходным числом, а справа от черты в строке с исходным числом ставят 0, если деление без остатка, и 1, если остаток есть. Деление повторяют до тех пор, пока делимое не станет меньше делителя. Считывание результата производится снизу вверх.

Пример. В десятичной системе число $A_{10} = 37$. Получить число А в двоичной системе счисления.



37	1
18	0
9	1
4	0
2	0
1	

$A_{10} = 37$

$A_2 = 100101$

Перевод *восьмеричных* и *шестнадцатеричных чисел* в двоичную систему осуществляется путем замены каждой цифры эквивалентной ей двоичной триадой (тройкой цифр) или тетрадой (четверкой цифр) соответственно.

Пример.

$537{,}1_8$ = 101 011 111, 001_2; $1A3, F_{16}$ = 1 1010 0011, 1111_2.

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

5 3 7 1 1 A 3 F

Чтобы перевести число из *двоичной системы* в восьмеричную или шестнадцатеричную, его нужно разбить влево и вправо от запятой на триады (для восьмеричной) или тетрады (для шестнадцатеричной) и каждую такую группу заменить соответствующей восьмеричной или шестнадцатеричной цифрой.

Например:

$10101001{,}10111_2$ = 10 101 001, 101 110_2 = 251, 56_8;

↓ ↓ ↓ ↓ ↓

2 5 1 5 6

$10101001{,}10111_2$ = 1010 1001, 1011 1000_2 = A9, $B8_{16}$.

↓ ↓ ↓ ↓

A 9 B 8

При переводе числа из *двоичной (восьмеричной, шестнадцатеричной)* системы счисления в десятичную надо это число представить в виде суммы степеней основания его системы счисления.

Пример.

Разряды 3 2 1 0 -1

Число $1\ 0\ 1\ 1,\ 1_2 = 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 11{,}5_{10}$

Разряды 2 1 0 -1

Число $2\ 7\ 6,\ 5_8 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 + 5 \cdot 8^{-1} = 190{,}625_{10}$

Разряды 2 1 0

Число $1\ F\ 3_{16} = 1 \cdot 16^2 + 15 \cdot 16^1 + 3 \cdot 16^0 = 499_{10}$

Контрольные вопросы

1. Что такое система счисления?

2. Какие система счисления используются для представления информации в компьютерах?

3. В чем отличие позиционной системы счисления от непозиционной?

4. Что называется основанием системы счисления?

5. Назовите порядок перевода чисел из десятичной в двоичную, восьмеричную и шестнадцатеричную системы счисления.

2. ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВЫХ УСТРОЙСТВ

2.1. Основы алгебры логики

Основой построения любого устройства, использующего цифровую информацию, являются элементы двух типов: логические и запоминающие. Логические элементы выполняют простейшие логические операции над цифровой информацией, а запоминающие – ее хранят.

Логическая операция состоит в преобразовании по определенным правилам входной информации в выходную. Сигналы на входах и выходах логических элементов обычно являются двоичными (бинарными), т. е. принимают лишь два значения, символически обозначаемые как 0 и 1. Поэтому их также называют двоичными переменными и обозначают буквами латинского алфавита (входные сигналы x_l, x_2, ..., x_n, а результат операции, т. е. выходной сигнал – y). Переменная x может принимать два значения: либо x = 1 (событие истинно), либо x = 0 (событие ложно). Эти переменные называются также булевыми по имени английского математика Дж. Булля, который в середине XIX века разработал основные положения алгебры логики.

Различные логические переменные могут быть связаны функциональными зависимостями. Например, $y = f(x_1, x_2)$ указывает на функциональную зависимость логической переменной y от логических переменных x_1 и x_2, называемых аргументами (или входными переменными).

2.2. Основные законы алгебры логики

В алгебре логики имеются четыре основных закона: переместительный (свойство коммутативности); сочетательный (свойство ассоциативности); распределительный (свойство дистрибутивности); инверсии (закон де Моргана).

Переместительный и сочетательный законы имеют место в обычной алгебре. Распределительного закона и закона инверсии в обычной алгебре нет.

Соотношения, отображающие основные законы алгебры логики, приведены в таблице 2.1.

Таблица 2.1

п/п	Закон	Логическое сложение	Логическое умножение
1	Переместительный	$x_1 \vee x_2 = x_2 \vee x_1$	$x_1 x_2 = x_2 x_1$
2	Сочетательный	$(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$	$(x_1 x_2)\, x_3 = x_1 (x_2 x_3)$
3	Распределительный	$(x_1 \vee x_2)\, x_3 = x_1 x_3 \vee x_2 x_3$	$x_1 x_2 \vee x_3 = (x_1 \vee x_3)(x_2 \vee x_3)$
4	Инверсии	$\overline{x_1 \vee x_2} = \overline{x}_1 \overline{x}_2$	$\overline{x_1 x_2} = \overline{x}_1 \vee \overline{x}_2$

Используя основные законы алгебры логики, можно составить ряд правил, которые применяются при анализе сложных логических выражений, приведения их к более простому и удобному виду (таблица 2.2).

Законам и правилам булевой алгебры присуще свойство симметрии. Все правила (кроме последнего) представлены парой соотношений. В каждой паре одно соотношение вытекает из другого заменой логического сложения логическим умножением и наоборот. Кроме того, все значения «0» заменяются на «1» и наоборот. Это свойство симметрии отражает принцип двойственности булевой алгебры.

Таблица 2.2

п/п	Правило	а	б
1	Инверсии	$\overline{0} = 1$	$\overline{1} = 0$
2	Неизменности	$x \vee 0 = x$	$x \cdot 1 = x$
3	Универсального и нулевого множества	$x \vee 1 = 1$	$x \cdot 0 = 0$
4	Повторения	$x \vee x = x$	$x \cdot x = x$
5	Дополнительности	$x \vee \overline{x} = 1$	$x \cdot \overline{x} = 0$
6	Склеивания	$x_1 x_2 \vee x_1 \overline{x}_2 = x_1$	$(x_1 \vee x_2)(x_1 \vee \overline{x}_2) = x_1$
7	Двойного отрицания	$\overline{\overline{x}} = x$	

Правило склеивания широко используется при минимизации логических функций с целью их упрощения.

2.3. Преобразование булевых выражений

Рассмотренные законы и правила используются для тождественных преобразований булевых выражений, описывающих логические функции. Булевое выражение представляет собой формулу, состоящую из логических констант и логических переменных, соединенных знаками логических операций. Как и в обычной алгебре для задания порядка выполнения логических выражений используются скобки. Примером булевого выражения для трех переменных служит формула

$$f(x_1, x_2, x_3) = (\overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2)(x_1 \vee x_3). \qquad (2.1)$$

Обычно считают, что операция логического умножения всегда предшествует операции логического сложения.

Преобразуем выражение (2.1), применив законы и правила алгебры логики.

$$(\overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2)(x_1 \vee x_3) = \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \cdot x_1 \vee \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot x_1 \vee$$
$$\vee x_1 \cdot \overline{x}_2 \cdot x_3 = \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \vee x_1 \cdot \overline{x}_2 \cdot x_3 = \overline{x}_2 \cdot x_3(\overline{x}_1 \vee x_1) \vee x_1 \cdot \overline{x}_2 =$$
$$= \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 = \overline{x}_2(x_3 \vee x_1). \qquad (2.2)$$

Полученное в результате тождественных преобразований выражение (2.2) значительно проще выражения (2.1). Процесс упрощения логического выражения, основанный на тождественных преобразованиях, называется минимизацией булевых выражений.

2.4. Дизъюнктивные нормальные формы

Для записи одной и той же функции алгебры логики можно использовать различные формы. Формы, которые представляют суммы элементарных произведений, называют дизъюнктивными нормальными формами (ДНФ).

Под элементарным понимается такое произведение, в котором сомножителями являются только отдельные переменные или их отрицания. Например, формула $f(x_1, x_2, x_3) = \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2$ содержит два элементарных произведения, каждое из которых состоит из двух сомножителей. Очевидно, одна и та же функция может быть представлена множеством различных ДНФ. Однако существуют такие виды ДНФ,

в которых функция может быть записана единственным образом. Такие формы называют совершенными дизъюнктивными нормальными формами (СДНФ).

СДНФ определяется как сумма элементарных произведений, в которых каждая переменная встречается ровно один раз либо с отрицанием, либо без него.

Для преобразования функции (2.2) в СДНФ необходимо дополнить каждое элементарное произведение недостающими переменными так, чтобы тождественность преобразования не была нарушена:

$$f(x_1, x_2, x_3) = \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 = (x_1 \vee \overline{x}_1) \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot (x_3 \vee \overline{x}_3).$$

После раскрытия скобок и приведения подобных членов, получим функцию, записанную в СДНФ:

$$\begin{aligned} f(x_1, x_2, x_3) &= x_1 \cdot \overline{x}_2 \cdot x_3 \vee \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot \overline{x}_3) = \\ &= x_1 \cdot \overline{x}_2 \cdot x_3 \vee \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot \overline{x}_3. \end{aligned} \qquad (2.3)$$

Здесь каждая переменная (или ее отрицание) содержится по одному разу в каждом элементарном произведении. Функция (2.3) обращается в логическую единицу при трех различных комбинациях значений входных переменных:

$x_1 = 1, x_2 = 0, x_3 = 1$ – первая комбинация;

$x_1 = 0, x_2 = 0, x_3 = 1$ – вторая комбинация;

$x_1 = 1, x_2 = 0, x_3 = 0$ – третья комбинация.

Для каждой комбинации соответствующее элементарное произведение равно единице, для всех остальных комбинаций входных переменных – нулю.

Таблица истинности такой функции (таблица 2.3) содержит три строки, в которых функция равна 1.

Таблица 2.3

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Каждой из этих строк соответствует по одной из рассмотренных выше комбинаций входных переменных, т. е. таблица истинности имеет столько строк, где функция обращается в 1, сколько элементарных произведений содержит ее СДНФ.

Чтобы написать СДНФ по таблице истинности, необходимо для всех комбинаций входных переменных, обращающих функцию в 1, записать элементарные произведения, инвертируя переменные, принимающие на данной комбинации нулевые значения, а все полученные элементарные произведения соединить знаками логического сложения.

Применив это правило, мы получим

$$f(x_1, x_2, x_3) = \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot \overline{x}_3 \vee x_1 \cdot \overline{x}_2 \cdot x_3 .$$

2.5. Минимизация логических функций

Рассмотренные тождественные преобразования позволяют существенно упростить выражения логических функций. Каждая логическая функция реализуется с помощью определенного набора устройств. Чем меньше элементов содержит выражение, тем проще схема, реализующая соответствующую ему логическую функцию. Поэтому значительный интерес представляет рассмотрение методов минимизации логических функций.

Наиболее распространенным является метод непосредственных тождественных преобразований. Этот метод, рассмотренный выше, состоит в последовательном применении к логической формуле законов и правил тождественных преобразований алгебры логики. Он пригоден для простых формул, когда последовательность преобразований очевидна. Наиболее часто этот метод применяется для окончательной минимизации выражений, полученных после минимизации их другими методами.

Рассмотрим применение метода непосредственных преобразований на примере минимизации функции трех переменных, заданных ее СДНФ:

$$y = \overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot \overline{x}_3 \vee x_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot x_2 \cdot \overline{x}_3 \vee x_1 \cdot x_2 \cdot x_3 . \qquad (2.4)$$

Объединим попарно первое и третье, второе и третье, а также четвертое и пятое элементарные произведения:

$$y = (\overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \cdot x_3) \vee (x_1 \cdot \overline{x}_2 \cdot \overline{x}_3 \vee x_1 \cdot \overline{x}_2 \cdot x_3) \vee (x_1 \cdot x_2 \cdot \overline{x}_3 \vee$$
$$\vee x_1 \cdot x_2 \cdot x_3) = \overline{x}_2 \cdot x_3 \cdot (\overline{x}_1 \vee x_1) \vee x_1 \cdot \overline{x}_2 \cdot (\overline{x}_3 \vee x_3) \vee x_1 \cdot x_2 \cdot (\overline{x}_3 \vee x_3) =$$
$$= \overline{x}_2 \cdot x_3 \vee x_1 \cdot \overline{x}_2 \vee x_1 \cdot x_2 = \overline{x}_2 \cdot x_3 \vee x_1(\overline{x}_2 \vee x_2) = \overline{x}_2 \cdot x_3 \vee x_1.$$

Эта формула гораздо проще исходной СДНФ. Здесь в формуле (2.4) в каждой паре объединяемые элементарные произведения различаются лишь одной переменной, которая входит в первое произведение с отрицанием, а во второе – без отрицания. Такие элементарные произведения называются соседними. К соседним произведениям применима операция склеивания, в результате которой уменьшается число суммируемых произведений и на единицу уменьшается число переменных.

2.6. Табличные методы минимизации. Карты Карно

Стремление к алгоритмизации поиска соседних элементарных произведений привело к разработке табличных методов минимизации логических функций. Одним из них является метод, основанный на использовании карт Карно.

Карты Карно – это графическое представление таблиц истинности логических функций. Они представляют собой таблицы, содержащие по 2^n прямоугольных ячеек, где n – число логических переменных.

На рис. 2.1, 2.2 и 2.3 приведены структуры карт Карно для функции двух, трех и четырех переменных, а в таблицах 2.4 и 2.5 представлены таблицы истинности для функции двух, трех переменных соответственно.

Таблица 2.4

x_1	x_2	$f(x_1, x_2)$
0	0	f(0, 0)
0	1	f(0, 1)
1	0	f(1, 0)
1	1	f(1, 1)

x_1 \ x_2	0	1
0	f(0,0)	f(0,1)
1	f(1,0)	f(1,1)

Рис. 2.1. Структура карты Карно для функции двух переменных

Таблица 2.5

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	f(0,0,0)
0	0	1	f(0,0,1)
0	1	0	f(0,1,0)
0	1	1	f(0,1,1)
1	0	0	f(1,0,0)
1	0	1	f(1,0,1)
1	1	0	f(1,1,0)
1	1	1	f(1,1,1)

x_1 \ x_2x_3	00	01	11	10
0	f(0,0,0)	f(0,0,1)	f(0,1,1)	f(0,1,0)
1	f(1,0,0)	f(1,0,1)	f(1,1,1)	f(1,1,0)

Рис. 2.2. Структура карты Карно для функции трех переменных

x_1x_2 \ x_3x_4	00	01	11	10
00	f(0, 0, 0, 0)	f(0, 0, 0, 1)	f(0, 0, 1, 1)	f(0, 0, 1, 0)
01	f(0, 1, 0, 0)	f(0, 1, 0, 1)	f(0, 1, 1, 1)	f(0, 1, 1, 0)
11	f(1, 1, 0, 0)	f(1, 1, 0, 1)	f(1, 1, 1, 1)	f(1, 1, 1, 0)
10	f(1, 0, 0, 0)	f(1, 0, 0, 1)	f(1, 0, 1, 1)	f(1, 0, 1, 0)

Рис. 2.3. Структура карты Карно для функции четырех переменных

Карта Карно размечается системой координат, соответствующих значениям входных переменных, например, верхняя строка карты для функции трех переменных соответствует нулевому значению переменной x_1, а нижняя – ее единичному значению. Каждый столбец этой карты характеризуется значениями двух переменных: x_2 и x_3. Комбинация цифр, которыми отмечается каждый столбец, показывает, для каких значений переменных x_3 и x_2 вычисляется функция, размещаемая в клетках этого

столбца. Так, в случае карты Карно для функции четырех переменных, функция, расположенная в ячейках столбца с координатами 01, вычисляется при значениях переменных $x_3 = 0$, $x_4 = 1$. Функция, расположенная в ячейке на пересечении этого столбца и строки с координатами 11, определяется при наборе входных переменных $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 1$.

Если на указанном наборе функция равна 1, то ее СДНФ обязательно содержит элементарное произведение $x_1 \cdot x_2 \cdot \overline{x}_3 \cdot x_4$, принимающее при этом наборе единичное значение. Таким образом ячейки карты Карно, представляющие функцию, содержат столько единиц, сколько элементарных произведений содержится в ее СДНФ, причем каждой единице соответствует одно из элементарных произведений.

Координаты строк и столбцов в карте Карно следуют не в естественном порядке возрастания двоичных кодов, а в порядке 00; 01; 11; 10. Изменение порядка следования наборов кодов сделано для того, чтобы соседние наборы, отличающиеся между собой лишь цифрой какого-либо одного разряда, были соседними в геометрическом смысле.

Рассмотрим таблицу истинности (таблица 2.6) и структуру карты Карно (рис. 2.4) для функции $f(x_1, x_2, x_3) = \overline{x}_1 \vee \overline{x}_2 \cdot x_3$.

Таблица 2.6

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

x_1 \ x_2x_3	00	01	11	10
0	1	1	1	1
1	0	1	0	0

Рис. 2.4. Структура карты Карно для функции

Ячейки, в которых функция принимает значение 1, заполняются единицами. Процесс минимизации заключается в формировании прямоугольников, содержащих по 2^k ячеек, где k – целое число. В прямоугольники объединяются соседние ячейки, которые соответствуют соседним элементарным произведениям. Например, на рис. 2.4 объединены ячейки с координатами 001 и 101. При объединении этих ячеек образовался прямоугольник, в котором x_1 изменяет свое значение. Следовательно, оно исчезает при склеивании соответствующих элементарных произведений $\overline{x}_1 \cdot \overline{x}_2 \cdot x_3$ и $x_1 \cdot \overline{x}_2 \cdot x_3$. Ячейки, расположенные в первой строке, содержат 1 и являются соседними. Поэтому они объединяются в прямоугольник, содержащий $2^2 = 4$ ячейки. Переменные x_2 и x_3 в пределах прямоугольника меняют свое значение, следовательно, они исчезнут из результирующего элементарного произведения. Переменная x_1 является неизменной и равной нулю. Таким образом, элементарное произведение, полученное в результате объединения ячеек первой строки, содержит лишь элемент $\overline{x}_1$. Это следует из того, что четырем ячейкам первой строки соответствует сумма четырех элементарных произведений:

$$\overline{x}_1 \cdot \overline{x}_2 \cdot \overline{x}_3 \vee x_1 \cdot \overline{x}_2 \cdot x_3 \vee \overline{x}_1 \cdot x_2 \cdot x_3 \vee \overline{x}_1 \cdot x_2 \cdot \overline{x}_3 =$$
$$= \overline{x}_1 \cdot \overline{x}_2 \cdot (\overline{x}_3 \vee x_3) \vee \overline{x}_1 \cdot x_2 \cdot (x_3 \vee \overline{x}_3) = \overline{x}_1 \cdot \overline{x}_2 \vee \overline{x}_1 \cdot x_2 = \overline{x}_1 \cdot (\overline{x}_2 \vee x_2) = \overline{x}_1.$$

Совокупность прямоугольников, покрывающих все единицы, называют покрытием. Одна и та же ячейка может покрываться несколько раз. Итак, можно сделать следующие выводы:

1. Формула, получающаяся в результате минимизации логической функции с помощью карт Карно, содержит сумму стольких элементарных произведений, сколько прямоугольников имеется в покрытии.

2. Чем больше ячеек в прямоугольнике, тем меньше переменных содержится в соответствующем ему элементарном произведении.

Несмотря на то, что карты Карно изображаются на плоскости, соседство ячеек устанавливается на поверхности тора или цилиндра. Верхняя и нижняя границы карты Карно как бы «склеиваются», образуя цилиндр. При склеивании боковых границ получается тороидальная поверхность. Поэтому в примере (рис. 2.5) ячейки с координатами 1011 и 0011 являются соседними и объединяются в прямоугольники. Действительно, указанным ячейкам соответствует сумма элементарных произведений

$$\overline{x}_1 \cdot \overline{x}_2 \cdot x_3 \cdot x_4 \vee x_1 \cdot \overline{x}_2 \cdot x_3 \cdot x_4 = \overline{x}_2 \cdot x_3 \cdot x_4 \cdot (x_1 \vee \overline{x}_1) = \overline{x}_2 \cdot x_3 \cdot x_4 .$$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	0	1	0
01	1	0	0	1
11	1	0	0	1
10	0	0	1	0

Рис. 2.5. Карта Карно для функции четырех переменных

Аналогично объединяются и остальные четыре единичные ячейки. В результате их объединения получаем:

$$\overline{x}_1 \cdot x_2 \cdot \overline{x}_3 \cdot \overline{x}_4 \vee x_1 \cdot x_2 \cdot \overline{x}_3 \cdot \overline{x}_4 \vee \overline{x}_1 \cdot x_2 \cdot x_3 \cdot \overline{x}_4 \vee x_1 \cdot x_2 \cdot x_3 \cdot \overline{x}_4 =$$
$$= x_2 \cdot \overline{x}_3 \cdot \overline{x}_4 \cdot (x_1 \vee \overline{x}_1) \vee x_2 \cdot x_3 \cdot \overline{x}_4 \cdot (x_1 \vee \overline{x}_1) = x_2 \cdot \overline{x}_4 (x_3 \vee \overline{x}_3) = x_2 \cdot \overline{x}_4 .$$

Поэтому окончательно $f(x_1, x_2, x_3, x_4) = \overline{x}_2 \cdot x_3 \cdot x_4 \vee x_2 \cdot \overline{x}_4$.

Рассмотренные примеры позволяют сформулировать последовательность действий, выполняемых при минимизации логических функций с использованием карт Карно.

1. Изображается таблица для n переменных и производится разметка ее сторон.

2. Ячейки таблицы, соответствующие набором переменных, обращающих функцию в 1, заполняются единицами, остальные ячейки – нулями.

3. Выбирается наилучшее покрытие таблицы правильными прямоугольниками. Наилучшим считается такое покрытие, которое образовано минимальным числом прямоугольников, а если таких вариантов несколько, то из них выбирается тот, который дает максимальную суммарную площадь прямоугольников.

Качество минимизации оценивается коэффициентом покрытия:

$$k = m/s,$$

где m – общее количество прямоугольников; s – суммарное количество покрытых ячеек. Покрытие считается тем лучше, чем меньше его коэффициент k.

2.7. Неполностью определенные логические функции

При рассмотрении двоично-десятичных кодов мы отметили, что из 16 возможных комбинаций используются только 10, а остальные комбинации запрещены и возникать не должны. Если каждому разряду поставить в соответствие двоичную переменную, то для двоично-десятичных кодов получим шесть запрещенных комбинаций переменных. Они приведены в табл. 2.7. Если функция имеет запрещенные наборы переменных, то ее значения на указанных наборах не определены и в таблице истинности отмечаются знаком *. Например, в таблице для трех переменных представлена функция (табл. 2.8), имеющая три запрещенных набора переменных.

Двоичные функции, значения которых определены не для всех наборов входных переменных, называются неполностью определенными. На карте Карно ячейки, соответствующие запрещенным наборам переменных, также отмечаются знаком * (рис. 2.6). При минимизации неполностью определенной функции ее следует доопределить, т. е. неопределенные значения ячеек карты Карно произвольным образом заменить единицами или нулями.

x_1 \ x_2x_3	00	01	11	10
0	*	1	1	*
1	0	*	1	0

Рис. 2.6. Карта Карно для функции трех переменных

На рис. 2.7 показана функция $f_1(x_1, x_2, x_3)$, все значения * которой заменены единицами. Доопределенная функция имеет вид $f_1(x_1, x_2, x_3) = \overline{x}_1 \vee x_3$ (не зависит от x_2).

x_1 \ x_2x_3	00	01	11	10
0	1	1	1	1
1	0	1	1	0

Рис. 2.7. Замена знаков * функции f1(x1,x2,x3) единицей

Таблица 2.7

Цифра	x_1	x_2	x_3	x_4	Набор
0	0	0	0	0	Разрешенный
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
-	1	0	1	0	Запрещенный
-	1	0	1	1	
-	1	1	0	0	
-	1	1	0	1	
-	1	1	1	0	
-	1	1	1	1	

Если крайние ячейки верхней строки карты Карно дополнить нулями (рис. 2.8), то получим функцию f_2, отличную от f_1: $f_2(x_1, x_2, x_3) = x_3$.

x_1 \ x_2x_3	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Рис. 2.8. Замена знаков * нулями в верхней строке

Таблица 2.8

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	*
0	0	1	1
0	1	0	*
0	1	1	1
1	0	0	0
1	0	1	*
1	1	0	0
1	1	1	1

Эти примеры показывают возможности упрощения формулы неполностью определенной функции при ее соответствующем доопределении.

Если функция имеет m запрещенных наборов переменных, то может быть выбран тот вариант, при котором формула минимизированной функции будет наиболее простой.

2.8. Логические элементы и логические операции

Одной из основных операций цифровой обработки информации является реализация функциональных зависимостей $y = f(x_1, x_2, ..., x_n)$, ставящих в соответствие каждой комбинации значений двоичных переменных $x_1, x_2, ..., x_n$ значение двоичной переменной y. Функция такого типа называется переключательной или логической. Переключательную функцию можно задать таблицей, в левой части которой перечисляются комбинации значений аргументов, а в правой – значения функции.

Переключательную функцию можно задать в аналитической форме, т. е. в виде некоторого выражения, описывающего последовательность элементарных операций над аргументами функций. Совокупность переключательных функций, определяющая набор элементарных операций, достаточных для реализации любой переключательной функции, называется базисом. В большинстве случаев необходимые логические преобразования двоичных сигналов выполняются на базе трех элементарных операций: логического сложения, логического умножения и инверсии.

Логическое сложение (дизъюнкция), либо операция ИЛИ обозначается $\vee$ или знаком «+»:

$$y = x_1 \vee x_2 \vee ... \vee x_n, = x_1 + x_2 + ... + x_n.$$

Читается так: у есть x_1 или x_2 или, ..., или x_n. Иначе говоря, у есть единица, если хотя бы одно из слагаемых равно единице.

Логическое умножение (конъюнкция), либо операция И обозначается символом $\wedge$ или знаком « · »:

$$y = x_1 \wedge x_2 \wedge ... \wedge x_n = x_1 \cdot x_2 \cdot ... \cdot x_n.$$

Читается так: у есть x_1 и x_2 и ... и x_n. Другими словами, у есть единица только тогда, когда все сомножители равны единице.

Логическое отрицание, называемое также инверсией либо операцией НЕ, обозначается чертой над переменной $y = \overline{x}$, читается так: у есть не х.

Перечисленные операции образуют так называемую булеву алгебру. Известно, что любую переключательную функцию можно представить аналитическим выражением в булевой алгебре, т. е. совокупность логических функций, состоящая из логических сложения, умножения и отрицания, является базисом.

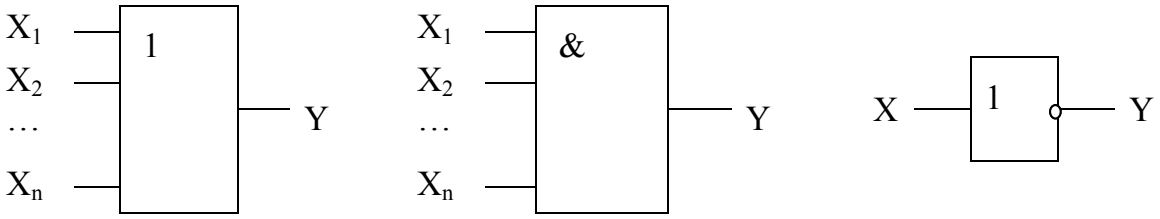
Правила выполнения логических операций над двоичными переменными для случая двух входных сигналов представлены в таблице 2.9, называемой таблицей истинности. Аналогично определяются операции ИЛИ, И для n входных переменных.

Логическим элементом (ЛЭ) или *логической схемой* называется устройство, реализующее заданную переключательную функцию. Обычно такое устройство имеет $n \geq 1$ входов и один выход.

Таблица 2.9

Операция ИЛИ			Операция И			Операция НЕ	
x_1	x_2	y	x_1	x_2	y	x	y
0	0	0	0	0	0	0	1
0	1	1	0	1	0	1	0
1	0	1	1	0	0		
1	1	1	1	1	1		

На функциональных и структурных схемах ЛЭ условно изображается прямоугольником, внутри которого записана реализуемая им логическая функция (рис. 2.9). Функция логического сложения условно обозначается символом 1, умножения – символом &, инверсия на выходе (входе) ЛЭ – кружком на выходе (входе) прямоугольника. Логический элемент, реализующий операцию ИЛИ (рис. 2.9, а), называется либо дизъюнктором, либо элементом ИЛИ, либо сборкой. Логический элемент, реализующий операцию И (рис. 2.9, б), называется либо конъюнктором, либо элементом И, либо схемой совпадения. Логический элемент, выполняющий операцию НЕ (рис. 2.9, в), называется инвертором.



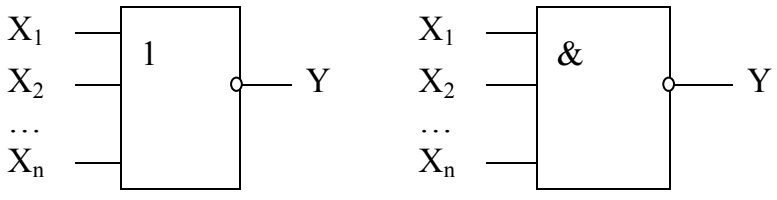
а б в

Рис. 2.9. Функциональное обозначение логических элементов:
а – элемент ИЛИ; б – элемент И; в – элемент НЕ

На практике широкое распространение получили комбинированные элементы, реализующие последовательно не одну, а две и более операции, например, элементы ИЛИ-НЕ (отрицание дизъюнкции), И-НЕ (отрицание конъюнкции). Логические функции, реализуемые этими элементами, записываются соответственно:

$y = \overline{x_1 + x_2 + ... + x_n}$, $y = \overline{x_1 \cdot x_2 \cdot ... \cdot x_n}$.

Условные обозначения этих элементов представлены на рис. 2.10 (а, б).



а б

Рис. 2.10. Функциональное обозначение логических элементов ИЛИ-НЕ (а) и И-НЕ (б)

Элементы ИЛИ-НЕ, И-НЕ являются универсальными, т. к. с помощью элементов одного из этих типов можно выполнить любую базисную функцию И, ИЛИ, НЕ.

2.9. Классификация логических элементов

Развитие микроэлектроники позволило в последние годы вести крупносерийное производство самых различных интегральных схем (ИС). Их разработка и производство ведется, как правило, в виде серий. Серия – это комплект ИС с различными логическими и электрическими характеристиками, имеющий единые схемотехническое и конструкторско-технологическое исполнения.

Существующие в настоящее время микросхемы могут быть классифицированы по многим признакам, но если выделить самое главное-различие топологии электрических схем основных (базовых) ЛЭ, то окажется, что все множество ИС может быть разделено на относительно небольшое число существенно различных систем.

Большинство современных ЛЭ относится к элементам потенциального типа, характерными чертами которых являются гальваническая связь между входом и выходом и возможность построения схемы без применения реактивных элементов или с использованием ограниченного числа конденсаторов малой емкости для вспомогательных целей.

В современных ЛЭ находят применение как биполярные, так и МОП-транзисторы. Возможности и основные свойства активных цепей таковы, что наиболее просто в схемном отношении реализуются операции И-НЕ, ИЛИ-НЕ. Логические элементы такого вида являются базовыми и в зависимости от конфигурации их схем выделяют следующие основные системы:

РТЛ – резисторно-транзисторная логика;

ДТЛ – диодно-транзисторная логика;

НСТЛ – транзисторная логика с непосредственной связью;

ТТЛ – транзисторно-транзисторная логика;

ТТЛШ – транзисторно-транзисторная логика с диодом Шоттки;

ЭСЛ – эмиттерно-связанная логика;

р-МОП – транзисторная логика на р-канальных МОП-транзисторах;

n-МОП – транзисторная логика на n-канальных МОП-транзисторах;

КМОП – транзисторная логика на комплементарных МОП-транзисторах.

В вычислительных устройствах применяется система положительных и отрицательных логических уровней. При положительной системе логических уровней

высокий уровень сигнала соответствует логической единице, а низкий уровень – логическому нулю. Этой системой удобно пользоваться в устройствах, выполненных на транзисторах n-p-n типа. Эту систему условно называют положительной логикой. При отрицательной системе логических уровней высокий уровень напряжения соответствует логическому нулю, а низкий – более отрицательный – логической единице. Этой системой удобно пользоваться в устройствах, выполненных на транзисторах p-n-p типа.

2.10. Основные характеристики логических элементов

Основными характеристиками логических элементов являются: статическая характеристика передачи, статическая помехоустойчивость, быстродействие, число входов или коэффициент объединения по входу, нагрузочная способность или коэффициент разветвления по выходу, потребляемая мощность.

Статическая характеристика передачи (рис. 2.11) представляет собой зависимость напряжения $U_{вых}$ на выходе ЛЭ от напряжения $U_{вх}$ на одном из его входов ($U_{вых} = f(U_{вх})$). При этом напряжения на других входах поддерживаются постоянными и соответствующими уровню логического нуля для схем ИЛИ-НЕ и уровню логической единицы для схем И-НЕ. В этом случае ЛЭ превращается в инвертор. К выходу исследуемого элемента подключается в качестве нагрузки однотипный элемент.

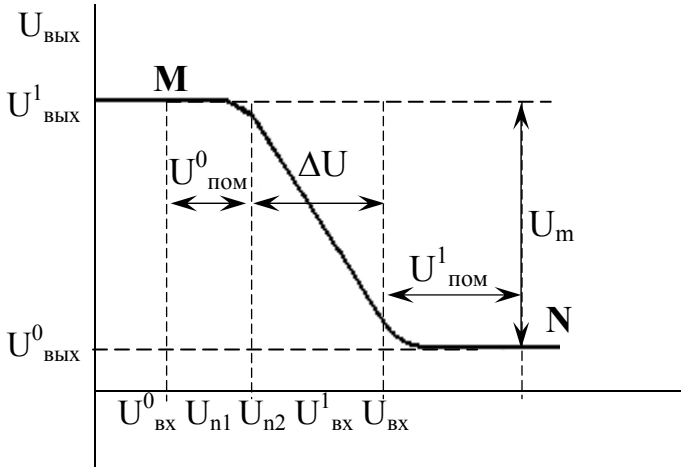


Рис. 2.11. Статическая характеристика передачи ЛЭ

Основные параметры статической характеристики передачи: уровни напряжения U^1 и U^0, равные соответственно логическим единице и нулю; логический размах

или перепад напряжения U_m; пороговые уровни U_{n1} и U_{n2} и ширина активной области ΔU. Отношение $U_m/\Delta U$ представляет собой значение среднего коэффициента передачи в активной области. Точки M и N характеризуют положение рабочей точки на статической характеристике передачи при подаче на вход ЛЭ уровней напряжения U^0 и U^1.

Статическая помехоустойчивость – определяется наибольшей величиной напряжений $U^0_{пом}$ и $U^1_{пом}$ (рис. 2.11), которые могут быть поданы на вход элемента относительно логических уровней 0 и 1 и не вызовут ложных переключений. Причиной таких помех могут быть паразитные падения напряжения на шинах питания. В практических схемах значение $U_{пом}$ колеблется от 0,1 до 0,3 В в элементах с низкой помехоустойчивостью и до 1,0 В – в элементах с высокой помехоустойчивостью.

Быстродействие в логических элементах определяется величиной задержки перепада напряжения при переходе его через ЛЭ. Эта задержка определяется наличием некоторого порога срабатывания элемента, инерционностью полупроводниковых приборов, влиянием паразитных емкостей. Она обычно измеряется на уровне, равном половине величины перепада и оказывается различной для положительного (t_3^+) и отрицательного (t_3^-) перепадов (рис. 2.12).

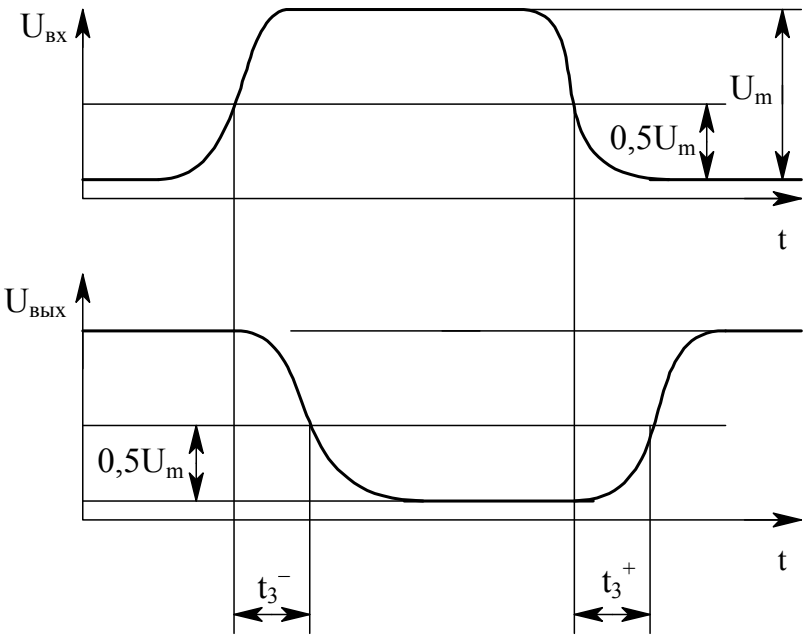


Рис. 2.12. Определение среднего времени задержки ЛЭ

Средней задержкой называют их полусумму: $t_{3.ср} = (t_3^+ + t_3^-)/2$.

Быстродействие логической схемы тем выше, чем меньше $t_{з.ср.}$. Все логические элементы по быстродействию можно разделить условно на 4 группы: сверхбыстродействующие – $t_{з.ср.} < 5$ нс; быстродействующие – $t_{з.ср.} = 5 – 10$ нс; среднебыстродействующие – $t_{з.ср.} = 11 – 15$ нс; медленнодействующие – $t_{з.ср.} > 15$ нс.

Коэффициент объединения по входу или число входов (m) логического элемента определяет максимальное количество входных сигналов, над которыми можно произвести операции ИЛИ и И, в зависимости от типа элемента m = 2 – 12.

Нагрузочная способность или *коэффициент разветвления по выходу* (n) характеризирует число входов аналогичных элементов, которые можно подключить к выходу данного элемента без нарушения его нормального функционирования. В зависимости от типа элемента n изменяется от 3 до 100.

Потребляемая мощность рассеивания в ЛЭ определяет не только его экономичность, но, что часто более важно, – степень его разогрева. Это ограничивает габариты элемента и всего устройства в целом. Уменьшение габаритов элемента и, следовательно, величины его охлаждающей поверхности при заданной величине его рассеивающей мощности приводит к росту температуры элемента сверх допустимого значения и к нарушению его работоспособности. Это обстоятельство оказывается особенно существенным в микроминиатюрных интегральных элементах. В сложном цифровом устройстве примерно половина входящих элементов в любой момент времени оказывается закрытой, а другая половина – открытой. Поэтому в качестве характеристики элемента пользуются понятием средней мощности $P_{ср}$, под которой понимается полусумма мощностей, рассеиваемых в двух статических состояниях элемента: закрытом и открытом. Величина $P_{ср}$ достигает сотен милливатт.

2.11. Резисторно-транзисторные логические элементы

Схема базового элемента РТЛ изображена на рис. 2.13. Ее основой является простейший ключ – инвертор, а операция ИЛИ реализуется входной сборкой резисторов R1, R2, R3. Схема относительно проста, но обладает рядом недостатков, основным из которых является сильная зависимость глубины насыщения транзистора T1 от числа логических единиц на входе, что приводит к ухудшению динамических характеристик транзистора. Эта же причина ограничивает функциональную сложность

ЛЭ – уменьшает коэффициент объединения по входу и коэффициент разветвления по выходу. Развитие интегральной схемотехники и технологии привело к тому, что в настоящее время система РТЛ практически не применяется.

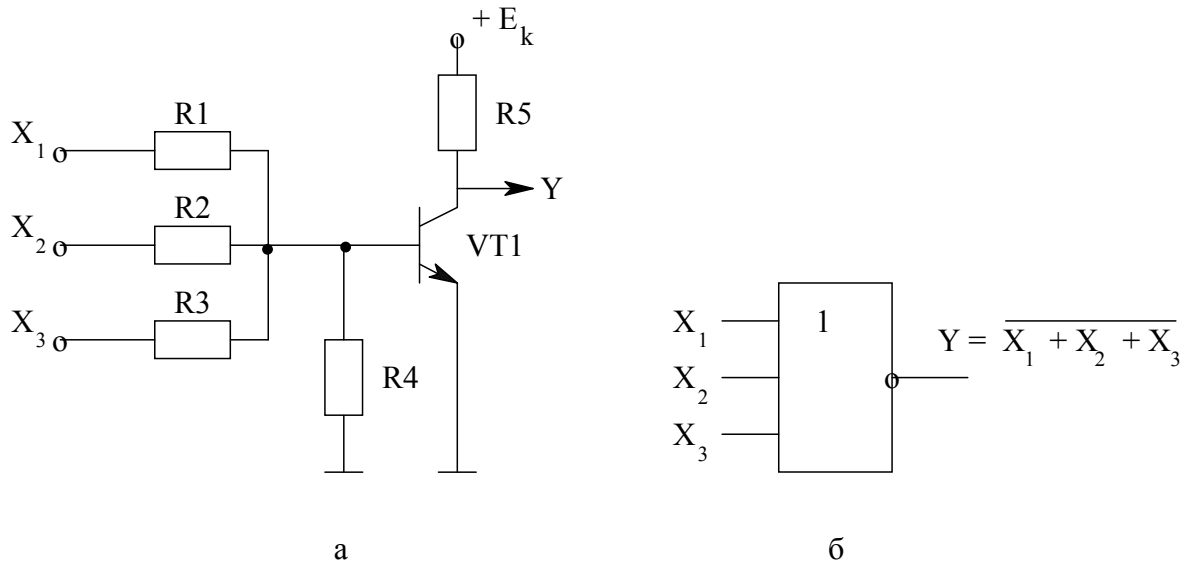


Рис. 2.13. Принципиальная схема (а) и функциональное обозначение (б) РТЛ элемента

Диодно-транзисторные логические элементы

Диодно-транзисторные логические элементы находят широкое применение в цифровых устройствах. Они имеют повышенную помехоустойчивость, среднее быстродействие и сравнительно большой коэффициент разветвления по выходу n. Основная схема ДТЛ-элемента, выполняющая логическую функцию И-НЕ, дана на рис. 2.14.

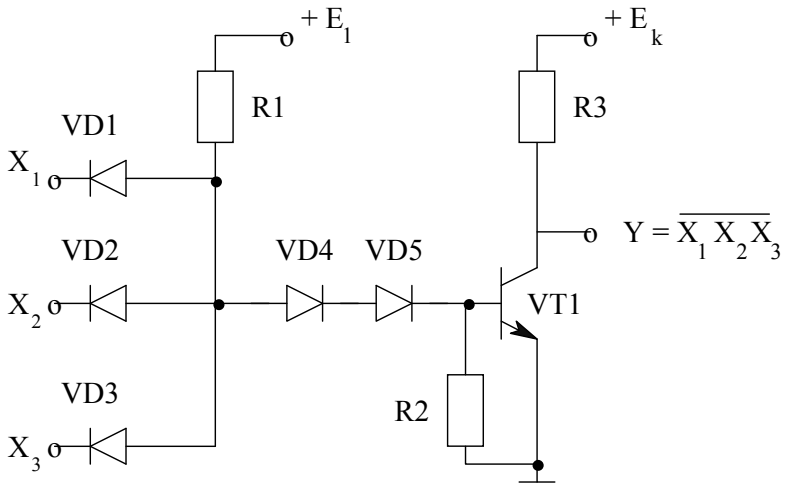


Рис. 2.14. Принципиальная схема ДТЛ-элемента

Диоды VD1 – VD3 на входе схемы совместно с резистором R1 выполняют логическую операцию И. Транзистор VT1 выполняет роль усилителя-инвертора. Диоды VD4, VD5 создают напряжение смещения, необходимое для совместимости уровней входного и выходного сигналов в схеме. В закрытом состоянии схемы резистор R2 задает ток через смещающие диоды, обеспечивая требуемое увеличение порога запирания транзистора VT1.

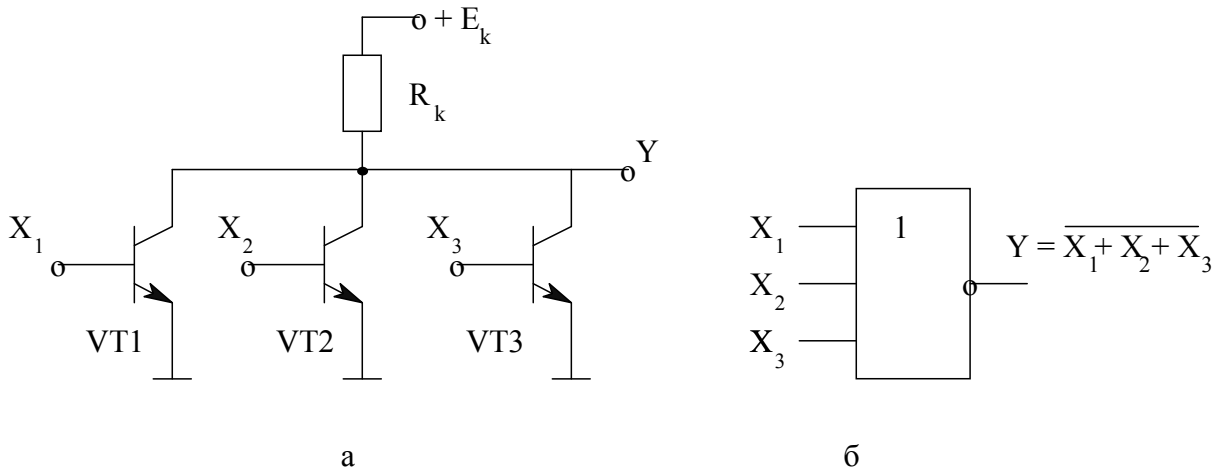
В открытом состоянии схемы транзистор VT1 находится в режиме насыщения и выходное напряжение схемы равно напряжению насыщения коллектора транзистора $U_{КН}$, а в закрытом состоянии оно близко к напряжению E_k. Схема задерживает сигнал при выключении. Это связано с рассасыванием избыточного заряда в базе, накопленного в режиме насыщения и наличием паразитных емкостей.

При подаче на все входы высокого уровня напряжения (логическая единица) входные диоды VD1 – VD3 смещаются в обратном направлении, и ток, протекающий через резистор R1, поступает в базу транзистора VT1 и вызывает его насыщение. Напряжение на выходе схемы равно напряжению нуля. Если на любом из входов схемы появляется низкий уровень напряжения, соответствующий уровню логического нуля, то соответствующий диод на входе открывается, диоды VD4, VD5 закрываются, и ток, протекающий через резистор R1, переключается из базовой цепи транзистора в цепь источника входного сигнала. Транзистор VT1 закрывается, выходное коллекторное напряжение схемы возрастает до напряжения E_k. Источник питания E_1 обеспечивает насыщение транзистора VT1.

2.12. Транзисторные элементы с непосредственной связью (НСТЛ)

Схема НСТЛ-элемента приведена на рис. 2.15.

Схема ЛЭ образована соединением коллекторов нескольких (на рисунке – трех) транзисторов. Здесь применена непосредственная связь выхода предыдущего каскада со входом последующего. Если на все входы поступают низкие уровни напряжения (логический нуль), то все три транзистора VT1 – VT3 закрыты и на выходе напряжение равно $+E_k$ (без учета подключения ЛЭ к нагрузке), что соответствует уровню логической единицы.



а б

Рис. 2.15. Принципиальная схема (а) и функциональное обозначение (б) НСТЛ-элемента

Если хотя бы на один вход поступает высокий уровень напряжения (логическая единица), то соответствующий транзистор открывается и насыщается и выходное напряжение снижается до $U_{кн}$, что соответствует логическому нулю.

При подаче логической единицы на большее число входов насыщается большее число транзисторов, но уровень выходного потенциала почти не изменяется и остается близким к нулю. Таким образом здесь реализуется операция ИЛИ-НЕ. При работе схемы на однотипные элементы уровень логической единицы определяется значением напряжения насыщения базы равным примерно 0,7–0,8 В кремниевых транзисторов, а значение напряжения насыщения коллектора равно 0,2–0,3 В. Такая малая разница уровней напряжения между логической единицей и нулем снижает помехоустойчивость данной схемы, но схема элемента НСТЛ отличается простотой и требует относительно небольшого числа компонентов, в результате чего эти элементы имеют особые преимущества для миниатюризации.

2.13. Транзисторно-транзисторные логические элементы

Схема ТТЛ ЛЭ, выполняющего логическую функцию И-НЕ, серии 155 представлена на рис. 2.16. Схема содержит входной каскад, реализующий функцию И, фазоинверсный каскад с источником тока и выходной каскад с активной нагрузкой.

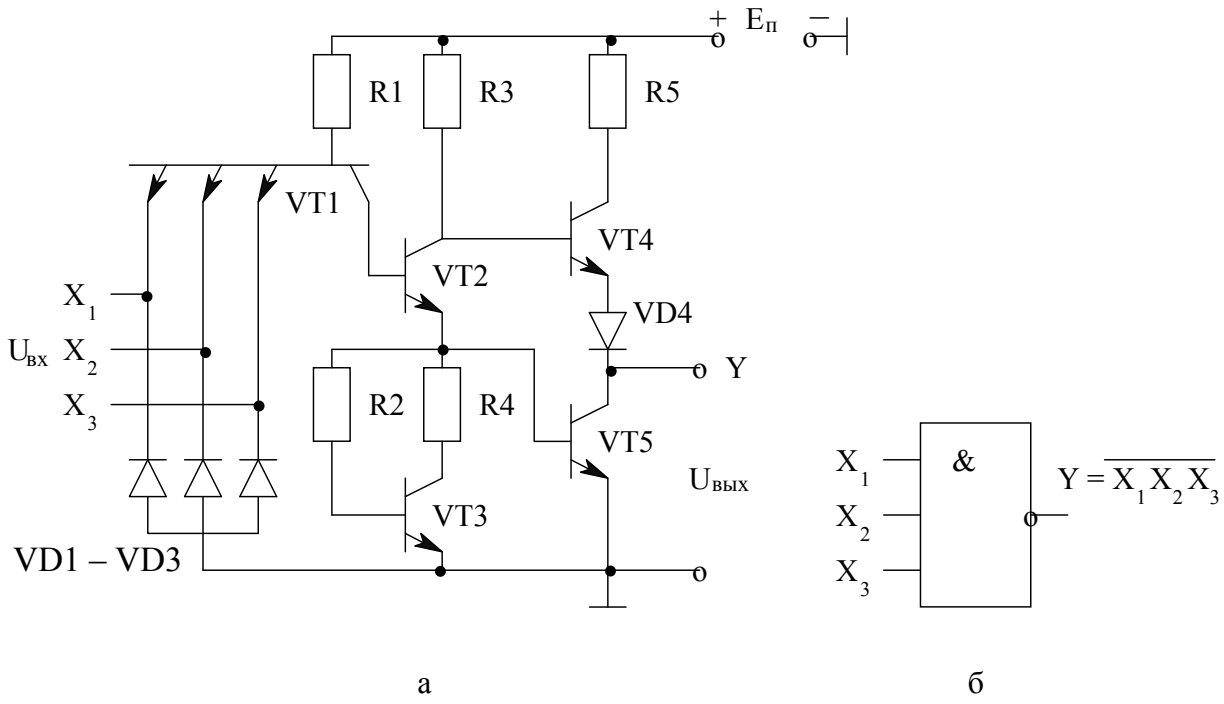


Рис. 2.16. Принципиальная схема (а) ТТЛ-элемента серии 155
и его функциональное обозначение (б)

Эмиттеры транзистора VT1 служат входами логического элемента и соединены с шиной нулевого потенциала через обратно-смещенные антизвонные диоды VD1 - VD3. Эти диоды служат своего рода демпфером, защищая от отрицательного входного напряжения транзистор VT1. При поступлении хотя бы на один эмиттер потенциала, близкого к нулю, через эмиттер начинает вытекать базовый ток транзистора VT1, создавая на входе ток логического нуля, ограничивающийся сопротивлением резистора R1. В таком режиме потенциал коллектора VT1 низок и транзисторы VT2 и VT5 закрыты. Транзистор VT4 открыт, обеспечивая на выходе уровень напряжения логической единицы:

$$U^1_{вых} = E_п - I_{бVT4} R3 - U_{VD4}.$$

Если на все эмиттеры транзистора VT1 подать напряжение $U^1_{вх}$, равное примерно напряжению питания $E_п$, то эмиттерные токи VT1 резко уменьшатся (входные токи логических единиц), а базовый ток уйдет в коллектор, создавая на базе транзистора VT2 высокий потенциал. Транзистор VT2 открывается, запирая при этом транзистор VT4 и отпирая транзистор VT5. Транзистор VT5 входит в насыщение, обеспечивая на выходе напряжение, равное примерно 0,4 В. Это напряжение есть напряжение логического нуля $U^0_{вых}$ (положительная логика).

Источник тока на VT3 и эмиттерный повторитель на VT5 способствуют улучшению передаточной характеристики логического элемента. При отпирании транзистора VT2 потенциал коллектора VT5 начинает падать (рис. 2.17), и в отсутствии источника тока, вместо которого может быть включен резистор, в точке 1 передаточной характеристики произошел бы излом, продолжение которого показано пунктиром.

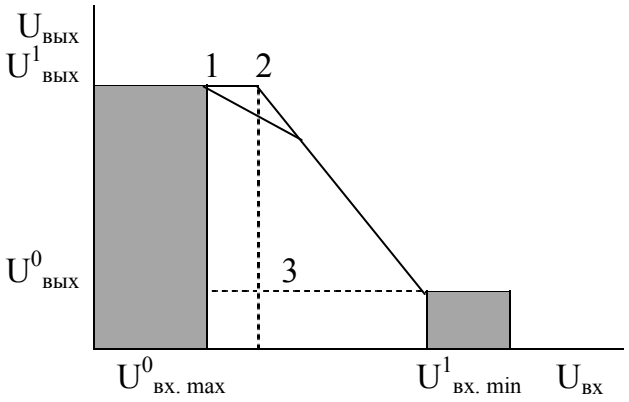
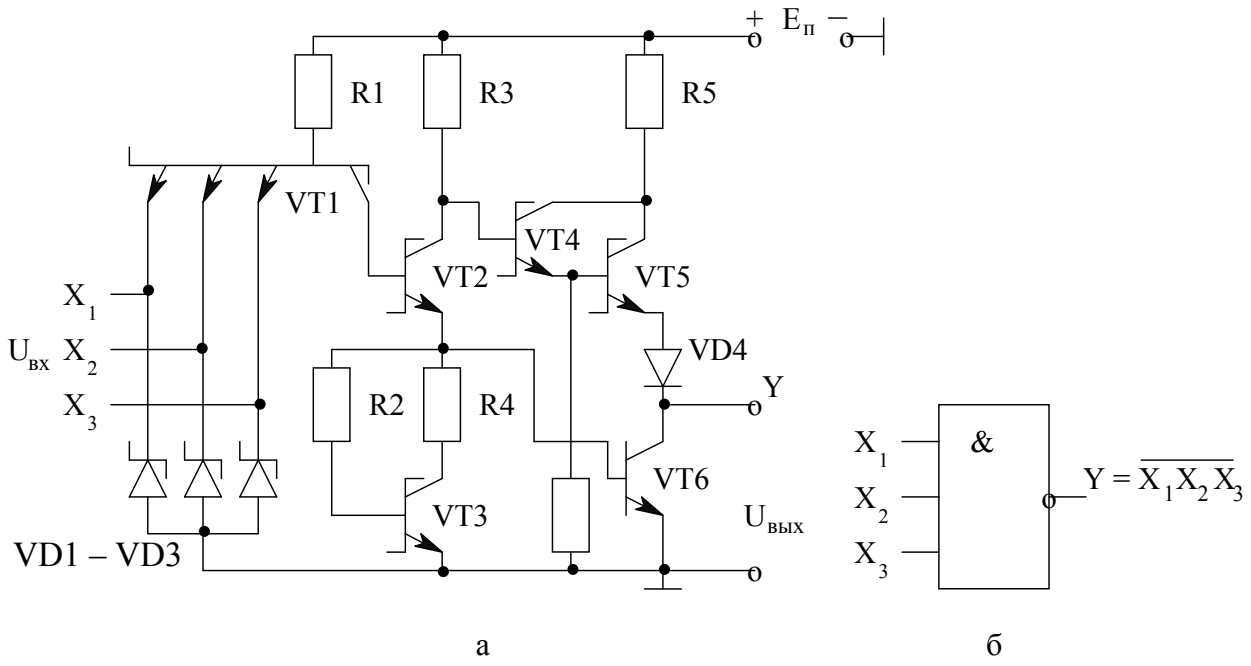


Рис. 2.17. Аппроксимированная передаточная характеристика логического элемента ТТЛ (заштрихованы области допустимых значений $U^0_{вх}$ и $U^1_{вх}$ в статическом режиме)

Положение точки излома определяется напряжением отпирания транзистора VT2, это максимально допустимое, т. е. пороговое напряжение логического нуля на входе $U^0_{вх.max} \approx 0{,}8$ В. Однако источник тока на транзисторе VT3 проявляет свойства нелинейного сопротивления и в начальный период повышения $U_{вх}$ ограничивает коллекторный ток транзистора VT2, сдерживая спад выходного напряжения $U^1_{вых}$. В то же время с повышением $U_{вх}$ медленно повышается и потенциал эмиттера VT2 и в точке 2 транзистор VT5 отпирается. В результате происходит резкое снижение выходного напряжения до уровня $U^0_{вых}$, когда $U_{вх}$ достигает значения минимально допустимого напряжения логической единицы $U^1_{вх.min} \approx 2{,}0$ В (точка 3 на рис. 2.17), переходные процессы заканчиваются, т. к. транзистор VT5 оказывается насыщенным и дальнейшее повышение $U_{вх}$ на потенциале коллектора практически не отражается.

Логические элементы серий 530, 531 явились результатом совершенствования микроэлектронной технологии, позволившей в начале 70-х годов начать изготовление в масштабах серийного производства выпрямляющих контактов металл-полупроводник, известных как переходы Шоттки (рис. 2.18).



а б

Рис. 2.18. Схема логического элемента И-НЕ серий 530, 531 ТТЛШ (а) и его функциональное обозначение (б)

Идея использования нелинейной отрицательной обратной связи для повышения быстродействия транзисторных ключей состоит в следующем. Известно, что время, затрачиваемое на формирование фронта выходного импульса, определяется рассасыванием инжектированных неосновных носителей, когда транзистор переходит из насыщения в область отсечки. Необходимо предотвратить вхождение транзистора в режим глубокого насыщения. Это может быть достигнуто путем приложения к участку база-коллектор запирающего напряжения.

Если между базой и коллектором включить диод Шоттки, подсоединив анодом к базе, то при отпирании транзистора на коллекторе в некоторый момент времени устанавливается потенциал, отпирающий диод Шоттки. Напряжение отпирания перехода Шоттки 0,4–0,5 В, т. е. меньше, чем падение на переходе база-коллектор и, следовательно, диод Шоттки откроется раньше, чем переход база-коллектор. Таким образом коллекторный переход оказывается запертым и режим насыщения исключается. Важным достоинством диодов Шоттки является то, что в них отсутствует инжекция неосновных носителей. В связи с этим при выключении не затрачивается время на рассасывание избыточного заряда и время их переключения составляет около 0,1 нс.

В серии ТТЛШ 1531, 1533 использованы транзисторы с диодом Шоттки с очень малым объемом коллекторной области, чем реализовано практически предельное быстродействие. На рис. 2.19 представлена схема ТТЛШ 1531 серии, реализующая операцию И-НЕ.

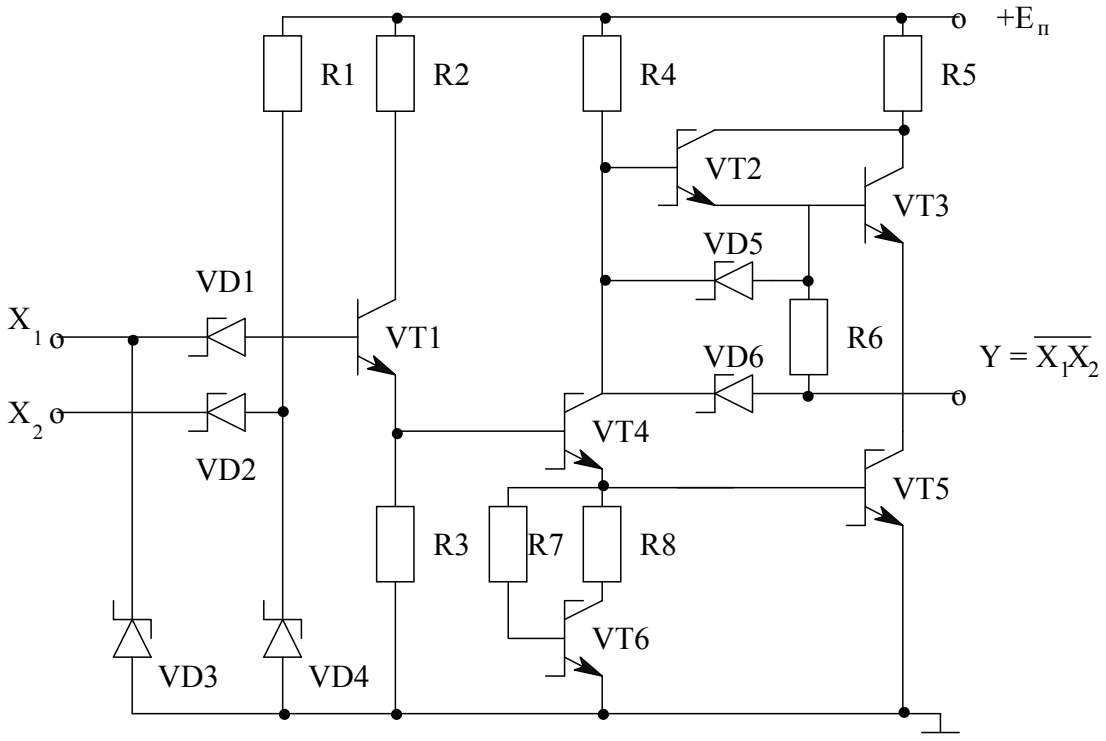


Рис. 2.19. Принципиальная схема элемента ТТЛШ серии 1531

Чтобы сохранить значительную нагрузочную способность элемента, входной ток низкого уровня уменьшен примерно в 10 раз. Для этого в схему ЛЭ добавлен после логической диодной матрицы эмиттерный повторитель на транзисторе VT1. Резисторы R1 и R3 фиксируют пороговое напряжение на уровне 1,5 В, что повышает помехоустойчивость ЛЭ.

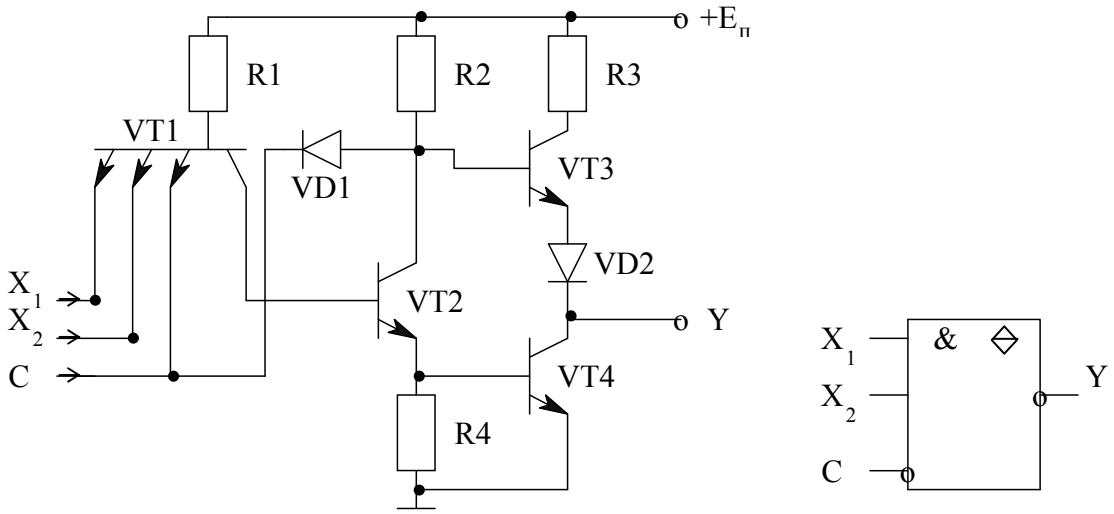
В таблице 2.10 приведены сравнительные характеристики различных серий ТТЛ-элементов по быстродействию и потребляемой мощности.

Таблица 2.10

Серия ТТЛ	К134	К155	К131	К555	К531	К1533	К1531
$t_{з.ср.}$, нс	33	9	6	9,5	3	4	3
$P_{ном}$, мВт	1	10	22	2	19	1,2	4

2.14. Логические элементы с тремя состояниями

Схема ЛЭ с тремя стабильными состояниями и ее функциональное изображение представлены на рис. 2.20.



а б

Рис. 2.20. Принципиальная схема (а)
и функциональное изображение ТТЛ-элемента с тремя состояниями (б)

Схемы с тремя состояниями имеют состояния по выходу: состояние логической единицы; состояние логического нуля; «обрыв». Они могут работать на общую шину, поэтому используются в качестве шинных формирователей. Схема работает следующим образом. Для управления транзисторами VT3 и VT4 введена дополнительная управляющая цепь, которая называется управлением (вход С). Этот вход через диод VD1 подключен к базе транзистора VT3, а также к эмиттеру транзистора VT1. Когда на вход С поступает напряжение, соответствующее уровню логического нуля, транзисторы VT3 и VT4 закрыты. Это соответствует третьему состоянию – «обрыв», т. е. логический элемент как бы отключается от выходной шины. Если на вход С подается высокий потенциал, то диод VD1 закрывается и транзистор VT3 будет работать как в обычной ТТЛ-схеме.

2.15. ТТЛ логические элементы с открытым коллектором

Схема ЛЭ с открытым коллектором приведена на рис. 2.21.

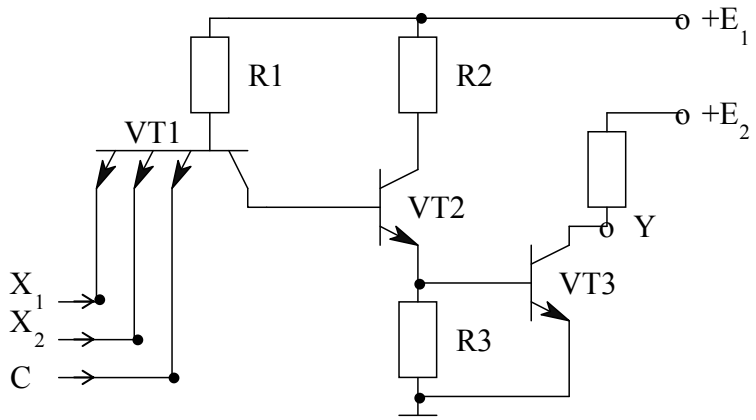


Рис. 2.21. Схема ТТЛ логического элемента с открытым коллектором

Эти элементы применяются в тех случаях, когда необходимо питать энергоемкую нагрузку. Выход с открытым коллектором позволяет иметь много параллельных выходов, что обеспечивает логическое сложение выходных функций схем при подаче напряжения питания E на группу соединенных вместе выходов через резистор соответствующей величины сопротивления, которая зависит от числа соединенных выходов и числа подключенных к ним входов последующих элементов. Кроме того, выход с открытым коллектором дает возможность в качестве нагрузки использовать лампы накаливания, светодиоды, обмотки реле, индикаторы и т. д.

2.16. ЭСЛ логические элементы

Логические элементы эммитерно-связанной логики (ЭСЛ) обладают наиболее высоким быстродействием из всех существующих ЛЭ, что обусловлено следующими факторами: открытые транзисторы работают в активном режиме, благодаря чему отсутствует задержка, связанная с рассасыванием избыточного заряда в транзисторах; внутренняя симметрия ЛЭ обеспечивает практическое отсутствие изменения потребляемого тока при переключении и связанных с ним всплесков напряжения в цепях питания.

Логические элементы ЭСЛ выполняются на базе переключателей тока (рис. 2.22). На базу транзистора VT0 подано напряжение E_0. Пусть $U_{вх1} = E_0$, при этом оба транзистора работают в активном режиме. Ток через резистор $R_э$ равен: $I_э = |E_0 - U_{бэо}| R_э$. Транзисторы выбирают идентичными, поэтому $i_{э1} = i_{э0} = I_э/2$.

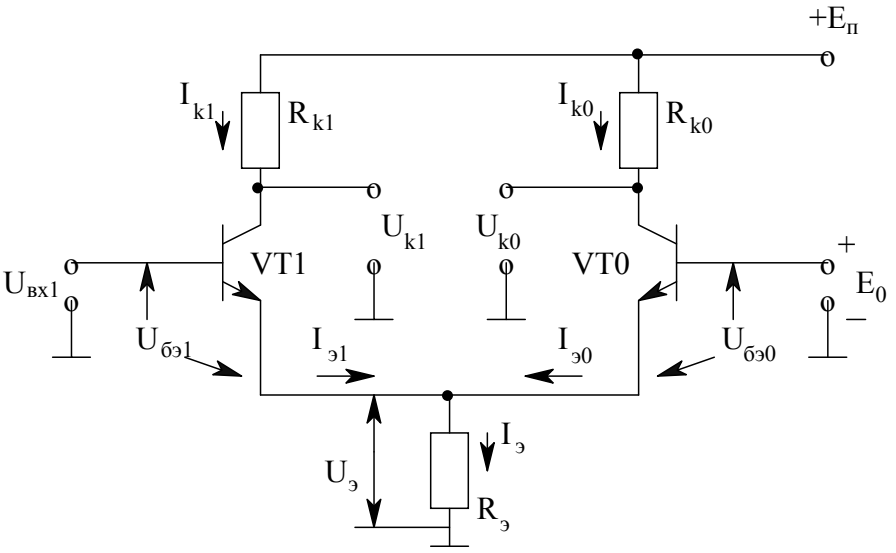


Рис. 2.22. Схема переключателя тока

$U_{k1} = U_{k0} = E_п - I_э R_{k1}/2$. Если входное управляющее напряжение $U_{вх1} > E_0$, то $i_{э1}$, $i_{к1}$ увеличиваются, напряжение U_{k1} падает, а напряжение $U_э$ растет, что приводит к уменьшению $U_{бэ0}$ и, следовательно, к уменьшению i_{k0} и к увеличению U_{k0}. Дальнейшее увеличение $U_{вх}$ приводит к полному запиранию транзистора VT0. При этом напряжение $U_{k0} = E_п$ (без учета неуправляемых тепловых токов), $U_{k1} = E_п - I_э R_k$. Ток $I_э \approx I_{к1}$ течет только через транзистор VT1.

При уменьшении напряжения $U_{вх}$ относительно E_0 уменьшается ток i_{k1}, растет напряжение U_{k1}, растет ток i_{k0}, уменьшается напряжение U_{k0} . При некотором $U_{вх} < U_0$ транзистор VT1 закрывается и ток $I_э$ течет только через транзистор VT0. В этом случае $U_{k0} = E_п - I_э R_{k0}$: $U_{k1} = E_п$. Таким образом, при изменении $U_{вх}$ на некоторое значение $2\Delta U$ симметрично относительно E_0, транзисторы переключаются (рис. 2.23).

Выходной перепад напряжения $U_{mвых} = I_э R_k$. Отметим, что высокий входной уровень $U^1_{вх}$ соответствует уровню логической единицы E^1, а низкий входной уровень $U^0_{вх}$ – уровню логического нуля E^0. Обычно выбирают $E_0 = 0{,}5\ (E^1 + E^0)$.

Схема ЭСЛ логического элемента представлена на рис. 2.24, в которой для реализации многовходовой переключательной функции входной транзистор заменяют группой параллельно соединенных транзисторов, а для согласования входных и выходных уровней логической единицы и нуля в схему включены эмиттерные повторители. При этом на выходе y формируется функция ИЛИ-НЕ, на выходе y_2 – функция ИЛИ.

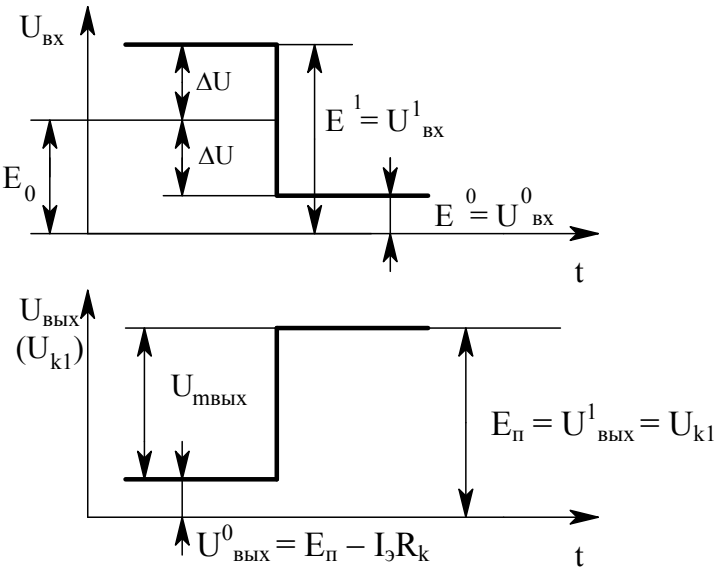


Рис. 2.23. Временная диаграмма состояния переключателя тока

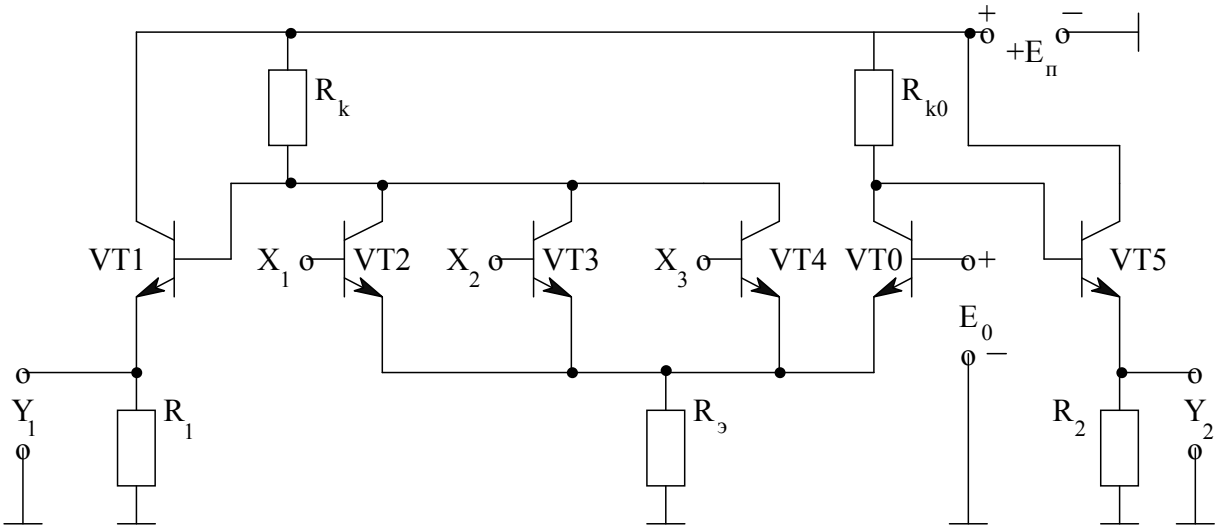


Рис. 2.24. Принципиальная схема ЭСЛ логического элемента

Имея высокое быстродействие, схемы ЭСЛ имеют и недостатки: низкая помехоустойчивость и большая потребляемая мощность, т. к. для достижения быстрого перезаряда паразитных емкостей в схеме используются резисторы с малым сопротивлением, через которые текут большие токи.

2.17. Логические элементы на МОП-транзисторах

Логические элементы на МОП-транзисторах изготавливаются по p-МОП, n-МОП, КМОП-технологиям. В первых используются p-канальные транзисторы, во вторых n-канальные транзисторы, а в третьих – используются комплементарные МОП-схемы, где в одной схеме применяются как p-канальные, так и n-канальные транзисторы. Схемы на МОП-транзисторах имеют меньшее быстродействие, чем схемы на биполярных транзисторах, что объясняется значительными емкостями между затвором, стоком, истоком и подложкой, на перезаряд которых требуется определенное время. Схемы на МОП-транзисторах потребляют малую мощность, имеют высокую нагрузочную способность и высокую помехоустойчивость, занимают малую площадь на поверхности кристалла, они технологичны и дешевы. Их применяют в тех случаях, где нужна большая степень интеграции и небольшое быстродействие. Особенность микросхем на МОП-структурах заключается в том, что в их схемах отсутствуют резисторы, а роль нелинейных резисторов выполняют соответствующим образом включенные транзисторы.

Логические элементы на p-канальных транзисторах. В основе всех логических схем на МОП-структурах лежит инвертор (рис. 2.25). При подаче на вход X напряжения –E (логическая единица), транзистор VT2 открывается, его сопротивление становится малым и на выходе мы получаем напряжение близкое к нулю (логический нуль). При подаче на вход напряжения логического нуля транзистор VT2 закрывается и на выходе мы имеем напряжение –E .

Для реализации функции И-НЕ ключевые транзисторы включают последовательно с нагрузочным транзистором VT1 (рис. 2.26). Если на каком-либо входе напряжение равно нулю (логический нуль), то соответствующий транзистор закрыт и напряжение на выходе равно –E (логическая единица).

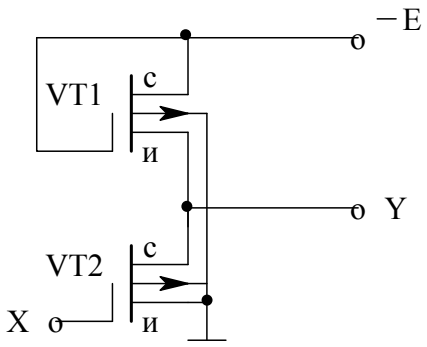


Рис. 2.25. Инвертор
на р-МОП-транзисторах

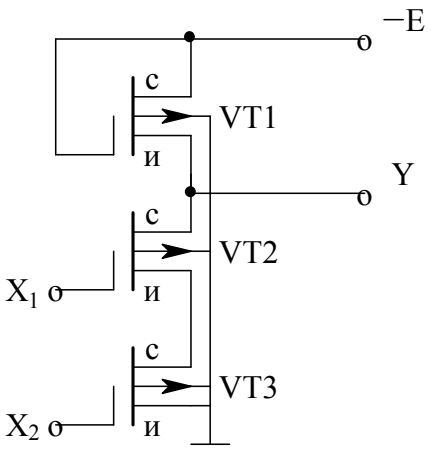


Рис. 2.26. Схема ЛЭ И-НЕ
на р-МОП-транзисторах

Если на все входы подать напряжение, соответствующее логической единице (–E), то транзисторы VT2 и VT3 будут открыты и иметь низкое сопротивление, выходное напряжение будет близко к нулю (логический нуль). Схема ИЛИ-НЕ образуется параллельным соединением переключательных транзисторов VT2 и VT3 и подсоединением их объединенных стоков к истоку нагрузочного транзистора VT1 (рис. 2.27).

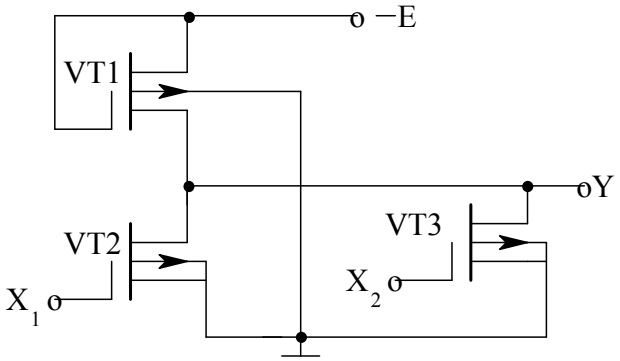


Рис. 2.27. Схема ЛЭ ИЛИ-НЕ на р-МОП-транзисторах

Если на любом из двух входов схемы напряжение соответствует логической единице (–E), то соответствующий транзистор открыт и имеет низкое сопротивление. При этом на выходе будет напряжение близкое к нулю (логический нуль). Если на все входы поданы логические нули, то все транзисторы будут закрыты и на выходе будет напряжение –E.

Логические схемы на КМОП-структурах. Преимущества ЛЭ на КМОП-структурах: малая потребляемая мощность, высокая помехоустойчивость, высокое быстродействие. Высокое быстродействие таких схем реализуется за счет того, что в цепях заряда и разряда паразитных емкостей схемы включены малые сопротивления открытых транзисторов. Логические схемы И-НЕ, ИЛИ-НЕ образуются параллельно-последовательным соединением ключевых транзисторов. На рис. 2.28 представлена схема, реализующая функцию ИЛИ-НЕ, использующая как p-канальные, так и n-канальные транзисторы.

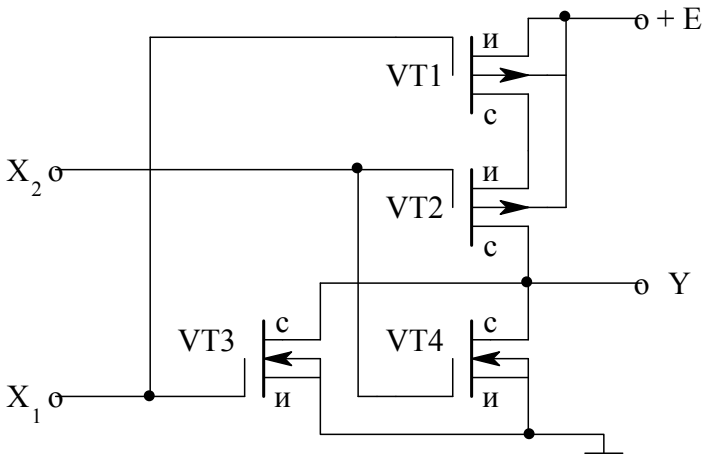


Рис. 2.28. Схема ЛЭ ИЛИ-НЕ на КМОП-структуре

Схема работает следующим образом. При подаче высокого уровня напряжения (+E) на вход x_1 транзистор VT1 закрывается, а транзистор VT3 открывается, на выходе будет напряжение близкое к нулю (логический нуль). То же самое будет и при подаче напряжения (+E) на вход x_2. При подаче на входы x_1 и x_2 логического нуля транзисторы VT3 и VT4 будут закрыты, а транзисторы VT1 и VT2 открыты. На выходе будет высокий уровень напряжения (+E) – логическая единица.

Логическая схема, реализующая функцию И-НЕ, представлена на рис. 2.29.

Если на вход x_1 подан уровень логического нуля (0 В), то транзистор VT1 будет открыт, а транзистор VT4 – закрыт. На выходе будет высокое напряжение (+E) – логическая единица. То же самое будет и при подаче нуля на вход x_2. При подаче на входы x_1 и x_2 логической единицы, транзисторы VT1 и VT2 будут закрыты, а транзисторы VT3, VT4 открыты – на выходе будет уровень логического нуля. В КМОП-логических схемах ток постоянно не протекает.

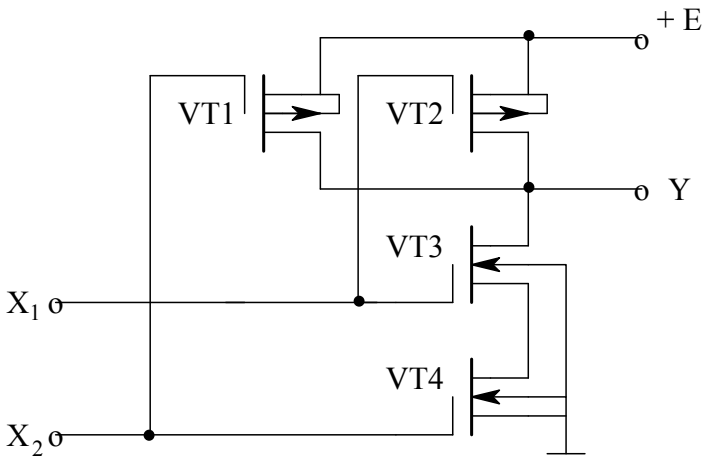


Рис. 2.29. Схема ЛЭ И-НЕ на КМОП-структуре

Коэффициент разветвления по выходу велик, поскольку эти схемы имеют очень большое входное сопротивление. КМОП-схемы могут работать в широком диапазоне питающих напряжений, что удобно при их совместной работе с ТТЛ схемами. Однако управлять КМОП-схемой ТТЛ схема не может, т.к. уровень напряжений логической единицы ТТЛ схемы недостаточен для КМОП-схемы. Согласование уровней ТТЛ и КМОП-схем достигается включением резистора между выходом ТТЛ схемы и источником питания (рис. 2.30).

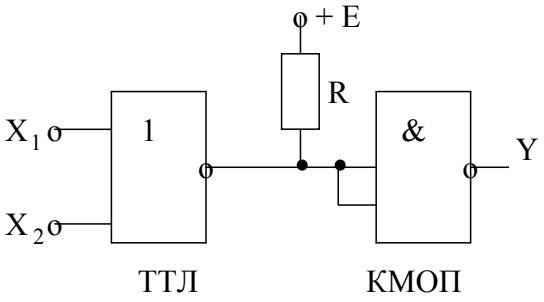


Рис. 2.30.Схема согласования ТТЛ и КМОП-логических элементов

Контрольные вопросы

1. Составить таблицу истинности переключательных функций:

$y_1 = x_1 + x_2$, $y_2 = x_1 \cdot x_2$.

2. Построить логические схемы на элементах И, ИЛИ, НЕ, реализующих переключательные функции $y_1 = x_1 \cdot x_2 + x_1 \cdot x_2$, $y_2 = x_1 + x_2 + x_1 \cdot x_2$.

3. Составить таблицу истинности переключательных функций:

$y_1 = x_1 + x_2$, $y_2 = x_1 \cdot x_2$.

4. Как по статической передаточной характеристике ЛЭ определить запас помехоустойчивости?

5. Перечислить факторы, обеспечивающие высокое быстродействие ЭСЛ-логических элементов. Объяснить назначение эмиттерных повторителей.

3. ЦИФРОВЫЕ КОМБИНАЦИОННЫЕ УСТРОЙСТВА

Цифровые комбинационные устройства (КУ) представляют собой логическую схему с m входами ($m \geq 1$) и n выходами ($n \geq 1$), у которой состояния выходов, т. е. информационные значения выходных сигналов, в данный момент времени определяются лишь состояниями входов в этот же момент времени. Синтез КУ выполняется на основе правил функционирования КУ, которые могут быть заданы словесно, в виде таблицы истинности, структурной формулой. Далее на основании правил алгебры логики или с помощью специальных методов (карт Карно и т. д.) производится минимизация структурной формулы КУ. Производят, если это необходимо, преобразование минимизированной структурной формулы к форме, содержащей лишь логические операции заданного базиса (И-НЕ, ИЛИ-НЕ). На основании структурной формулы составляют функциональную и принципиальную схемы КУ. Рассмотрим несколько примеров реализации КУ.

3.1. Устройство равнозначности

Устройство равнозначности (УР) — это логическая схема с двумя входами X_1 и X_2 и выходом Y, условия работы которой таковы: сигнал Y= 1 только при совпадении информационных значений входных сигналов. В таблице 3.1 приведена таблица истинности УР.

Структурная формула схемы для наборов, где Y = 1, имеет вид

$$Y = X_1X_2 + \overline{X}_1\overline{X}_2 .$$

Функциональная схема УР приведена на рисунке 3.1. Она содержит пять логических элементов — два инвертора, два элемента И и один элемент ИЛИ.

Таблица 3.1

X_1	X_2	Y
0	0	1
0	1	0
1	0	0
1	1	1

На основе правила де Моргана УР можно выполнить в базисе И-НЕ:

$$Y = X_1X_2 + \overline{X}_1\overline{X}_2 = \overline{\overline{X_1X_2}\,\overline{\overline{X}_1\overline{X}_2}}.$$

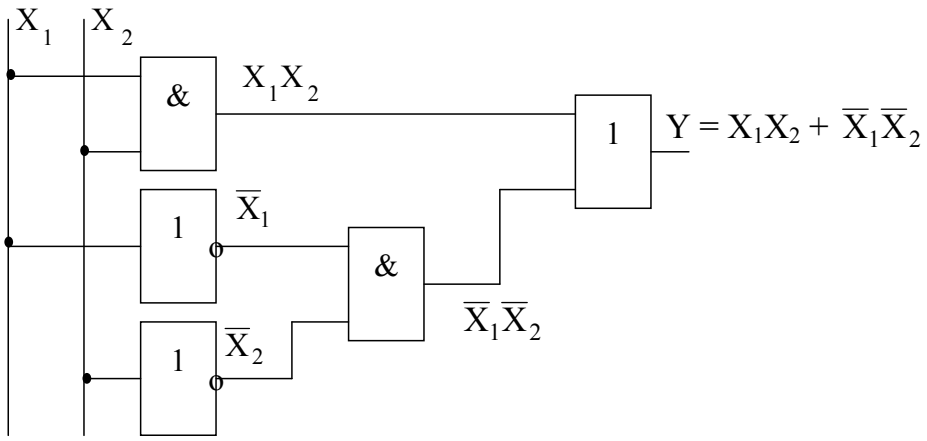


Рис. 3.1. Функциональная схема устройства равнозначности

Эту структурную формулу можно реализовать на пяти элементах И-НЕ (рис. 3.2).

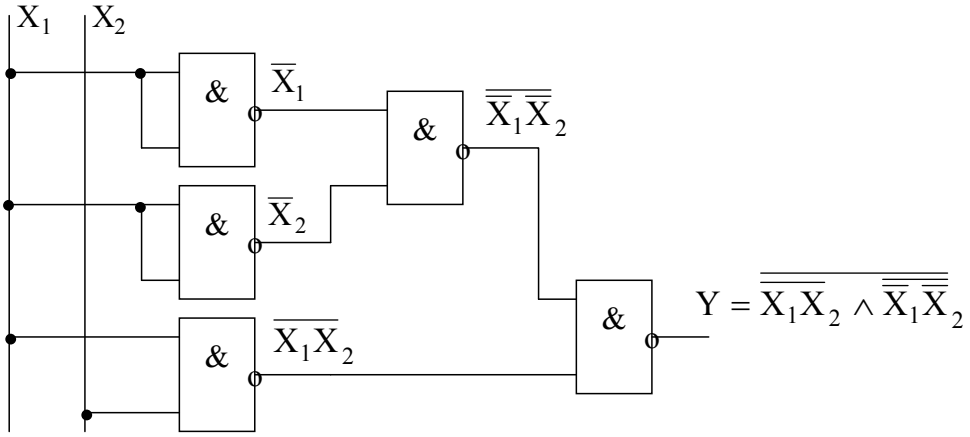


Рис. 3.2. Функциональная схема УР на элементах И-НЕ

Устройства равнозначности применяются в основном в схемах сравнения цифровых кодов.

3.2. Устройство неравнозначности

Устройство неравнозначности — это устройство с двумя входами X_1 и X_2 и выходом Y, реализующее логическую функцию, называемую «исключающее ИЛИ»: Y = l только при несовпадении информативных значений входных сигналов. Таблица истинности устройства неравнозначности представлена в таблице 3.2.

Таблица 3.2

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Согласно таблице истинности можно записать структурную формулу

$$Y = X_1\overline{X}_2 + \overline{X}_1X_2 = X_1 \oplus X_2.$$

Символ $\oplus$ означает операцию «исключающее ИЛИ». Устройство может быть реализовано схемой, содержащей пять логических элементов: два инвертора, два элемента И и один элемент ИЛИ (рис. 3.3).

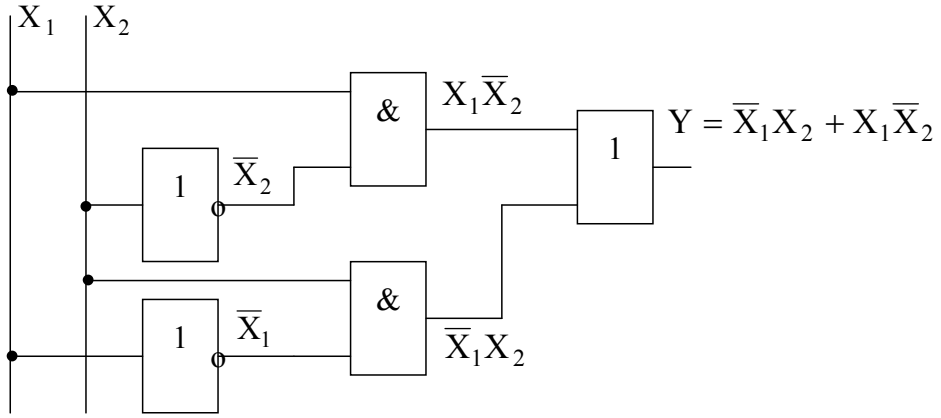


Рис. 3.3. Функциональная схема устройства неравнозначности

Устройства неравнозначности являются базовыми для построения комбинационных полусумматоров и сумматоров.

3.3. Комбинационные сумматоры

Рассмотрим логическую схему полусумматора, предназначенную для суммирования двух одноразрядных двоичных чисел. Логическая функция полусумматора задается следующей таблицей истинности (табл. 3.3).

Таблица 3.3

X_1	X_2	S'	P'
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Здесь X_1 и X_2 — переменные, S' — частичная сумма (сумма по модулю два), P' — перенос в старший разряд. Из таблицы истинности видно, что

$S' = X_1 \oplus X_2$, $P' = X_1 X_2$,

т. е. полусумматор имеет два выхода (S', P') и реализуется с помощью устройства неравнозначности и схемы И. Функциональная схема полусумматора представлена на рис. 3.4 (а, б).

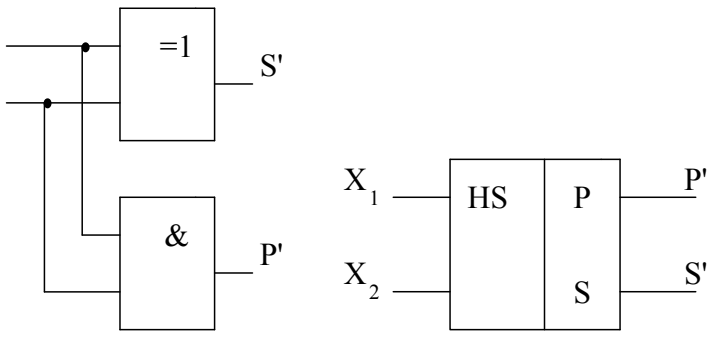


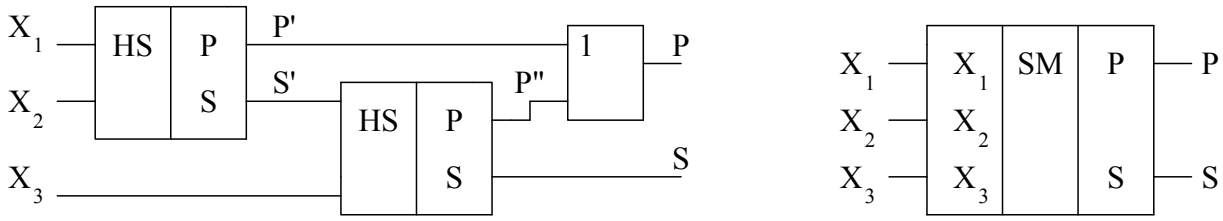
Рис. 3.4. Функциональная схема полусумматора (а)
и его условное графическое обозначение (б)

Два одноразрядных полусумматора образуют одноразрядный сумматор. Сумматор — логическая схема, имеющая три входа (входы разрядов X_1 и X_2

суммируемых чисел и вход X_3 для возможного переноса, образованного при сложении младших разрядов). Сумматор может быть образован с помощью двух полусумматоров и элемента ИЛИ. Таблица истинности (табл. 3.4) и функциональная схема сумматора показаны на рис. 3.5. В суммирующем устройстве последовательного действия для суммирования многоразрядных чисел перенос P подается на вход X_3 с задержкой на один такт сложения. Для ускорения процесса сложения двух n-разрядных чисел можно построить суммирующее устройство параллельного действия, в которых используются n сумматоров одноразрядных чисел. В настоящее время одно-, двух- и четырехразрядные двоичные сумматоры выпускаются в виде отдельных ИМС (К155ИМ1, К155ИМ2, К155ИМ3).

Таблица 3.4

X_1	X_2	X_3	S	P
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



а б

Рис. 3.5. Функциональная схема (а)
и условное графическое обозначение сумматора (б)

3.4. Дешифраторы

Дешифратором называется комбинационное устройство, имеющее несколько входов и выходов, у которого каждой комбинации входных сигналов соответствует активное значение только одного определенного выходного сигнала. Полный дешифратор с m входами имеет 2^m выходов. Таблица истинности трехвходового полного дешифратора с единичным активным значением выходных сигналов Y представлена в таблице 3.5.

Таблица 3.5

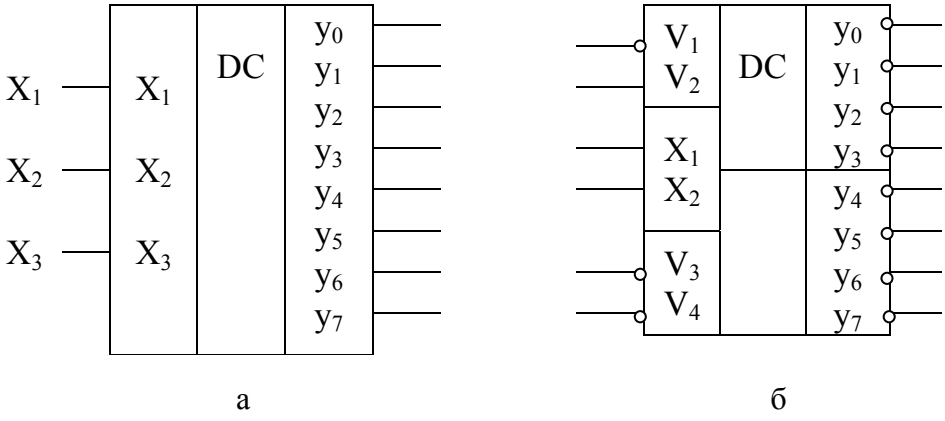
X_3	X_2	X_1	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

На практике часто используются неполные дешифраторы, предусматривающие декодирование только отдельных входных сигналов. Если задачу синтеза соответствующего КУ с тремя входами и восемью выходами рассматривать как синтез восьми одновходовых КУ, то для каждой из булевых функций Y можно записать структурную формулу:

$$Y_0 = \overline{X}_3\overline{X}_2\overline{X}_1,\ \ Y_1 = \overline{X}_3\overline{X}_2X_1,\ Y_2 = \overline{X}_3X_2\overline{X}_1,\ \ Y_3 = \overline{X}_3X_2X_1,$$
$$Y_4 = X_3\overline{X}_2\overline{X}_1,\ \ Y_5 = X_3\overline{X}_2X_1,\ Y_6 = X_3X_2\overline{X}_1,\ \ Y_7 = X_3X_2X_1. \qquad (3.1)$$

Условное графическое обозначение дешифратора, соответствующего табл. 3.5, представлено на рис. 3.6, а. В интегральном исполнении выпускаются различные структуры дешифраторов, в которых используются 2, 3, 4 входа. В одном корпусе может быть несколько дешифраторов.

Для увеличения функциональных возможностей устройств часто предусматривается использование нескольких сигналов управления. На рис. 3.6, б представлено графическое изображение микросхемы К155ИД4, содержащей сдвоенный двухвходовой дешифратор с активными нулевыми выходными сигналами.



а б

Рис. 3.6. Условное графическое обозначение дешифратора (а)
и условное графическое обозначение ИМС К155ИД4 (б)

На выходах обоих дешифраторов формируются сигналы в соответствии с комбинациями входных сигналов X_1 и X_2. Синхронизация процесса формирования выходных сигналов Y_0–Y_3 для каждого дешифратора задается комбинациями управляющих сигналов V. Для верхнего дешифратора разрешает формирование выходных сигналов комбинация $\overline{V}_1\, V_2$, для нижнего – $\overline{V}_3\, \overline{V}_4$. При отсутствии разрешающих комбинаций на каждом выходе Y устанавливается единичное значение сигнала. Введение такого управления расширяет возможности микросхемы при построении более сложных устройств, например, дешифраторов с увеличенным числом входов и выходов.

Переключательную функцию (формула 3.1) можно реализовать на отдельном элементе И, в этом случае мы получим наиболее простой по структуре дешифратор, который можно реализовать также на элементах И-НЕ (рис. 3.7).

Достоинство линейного дешифратора – высокое быстродействие; недостаток – большое количество требуемых входов логических элементов, равное количеству разрядов дешифрируемого слова. Кроме того, ограниченная нагрузочная способность ИМС ограничивает разрядность дешифрируемых слов до 4–5. Поэтому широко применяются пирамидальные дешифраторы, характерным отличием которых от линейных является использование только двухвходовых элементов.

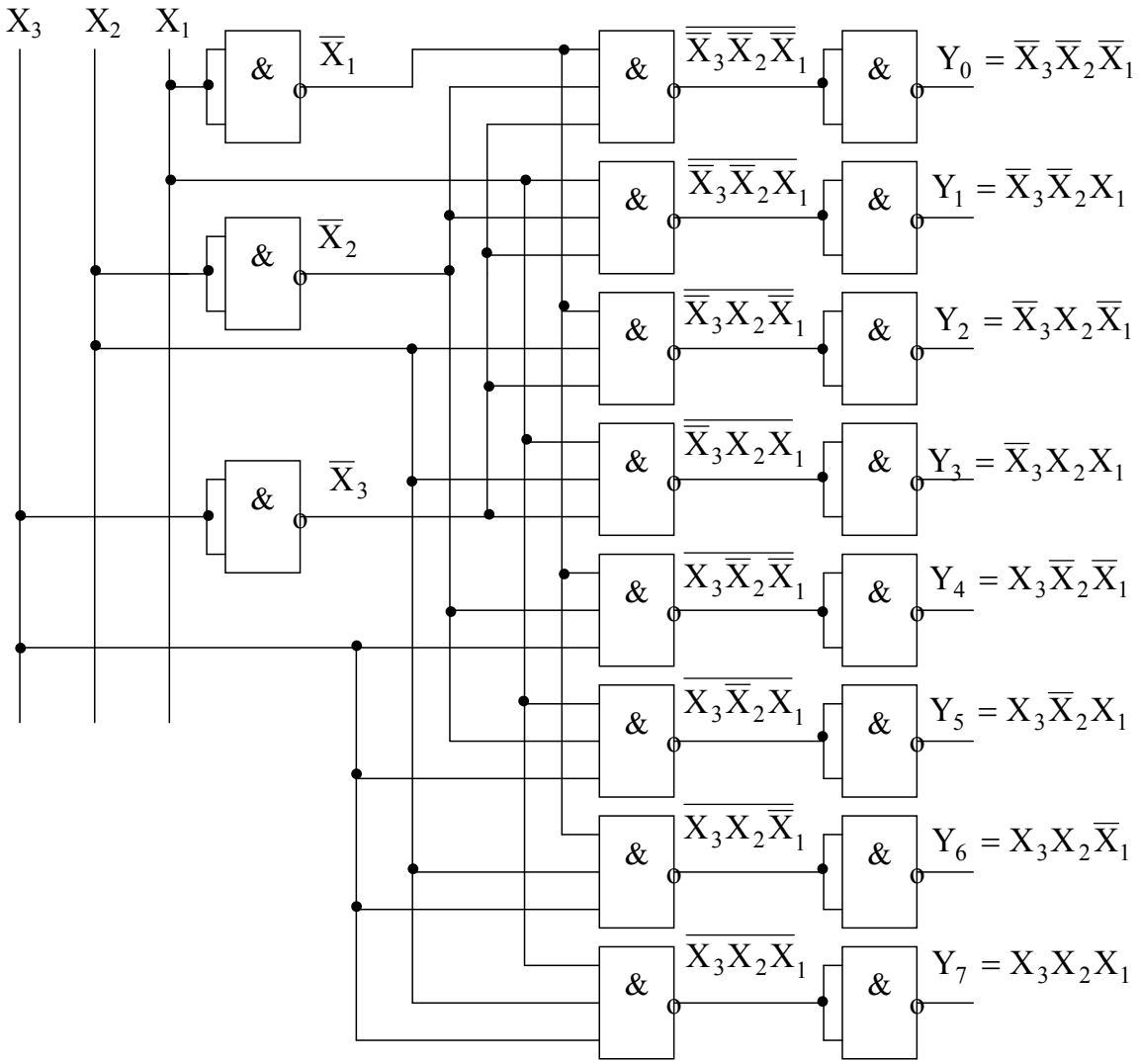


Рис. 3.7. Принципиальная схема линейного дешифратора на элементах И-НЕ

На рис. 3.8 представлена схема пирамидального дешифратора на 3 входа, выполненная на двухвходовых элементах И-НЕ. В пирамидальных дешифраторах требуется большее количество логических элементов, но по числу корпусов ИС пирамидальная схема может оказаться эквивалентной линейной, т. к. 2-входовых ЛЭ в одном корпусе размещается больше.

Третий вид структуры дешифратора — каскадное соединение дешифраторов. На рис. 3.9 показана группа из пяти дешифраторов, соединенных последовательно в два каскада. Все дешифраторы одинаковы и построены по структуре линейного дешифратора с инверсными выходами и дополнительным стробируемым входом С. На первый каскад, т. е. на входы DC_4, поданы старшие разряды дешифрируемого числа X_3, X_4. При этом DC определяет, в какой четверке из возможных 16 вариантов находится избранный выход.

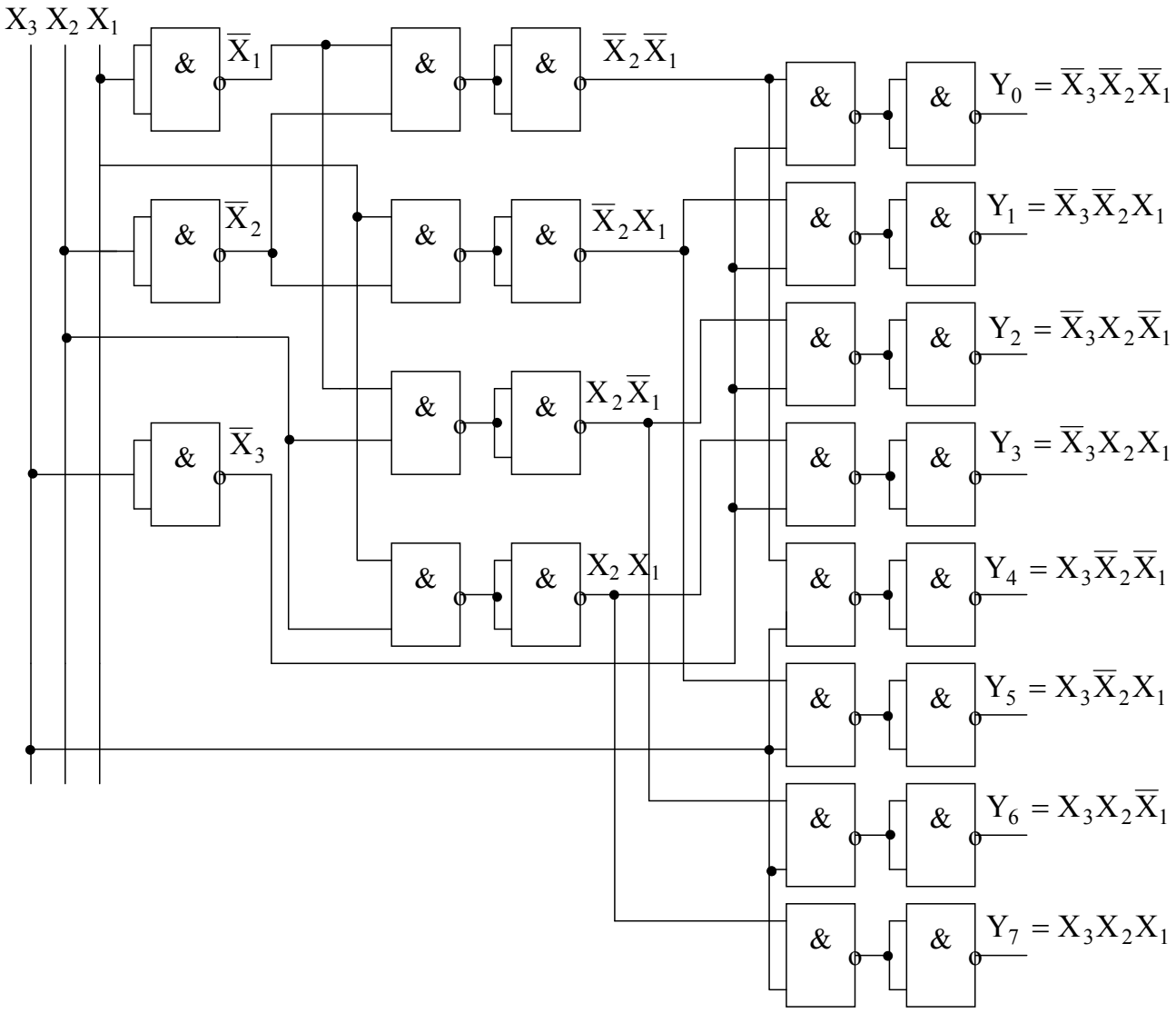


Рис. 3.8. Принципиальная схема пирамидального дешифратора

Дешифратор «открывает» соответствующий дешифратор второго каскада, подавая на инверсный вход С нулевой сигнал, а поскольку на прочих выходах DC формируется 1, все остальные дешифраторы второго каскада закрыты. На кодовые входы дешифраторов второго каскада поступают младшие разряды дешифрируемого кода. Причем работает единственный незакрытый дешифратор второго каскада, возбуждая в результате один из 16 выходов, соответствующий коду, поданному на вход.

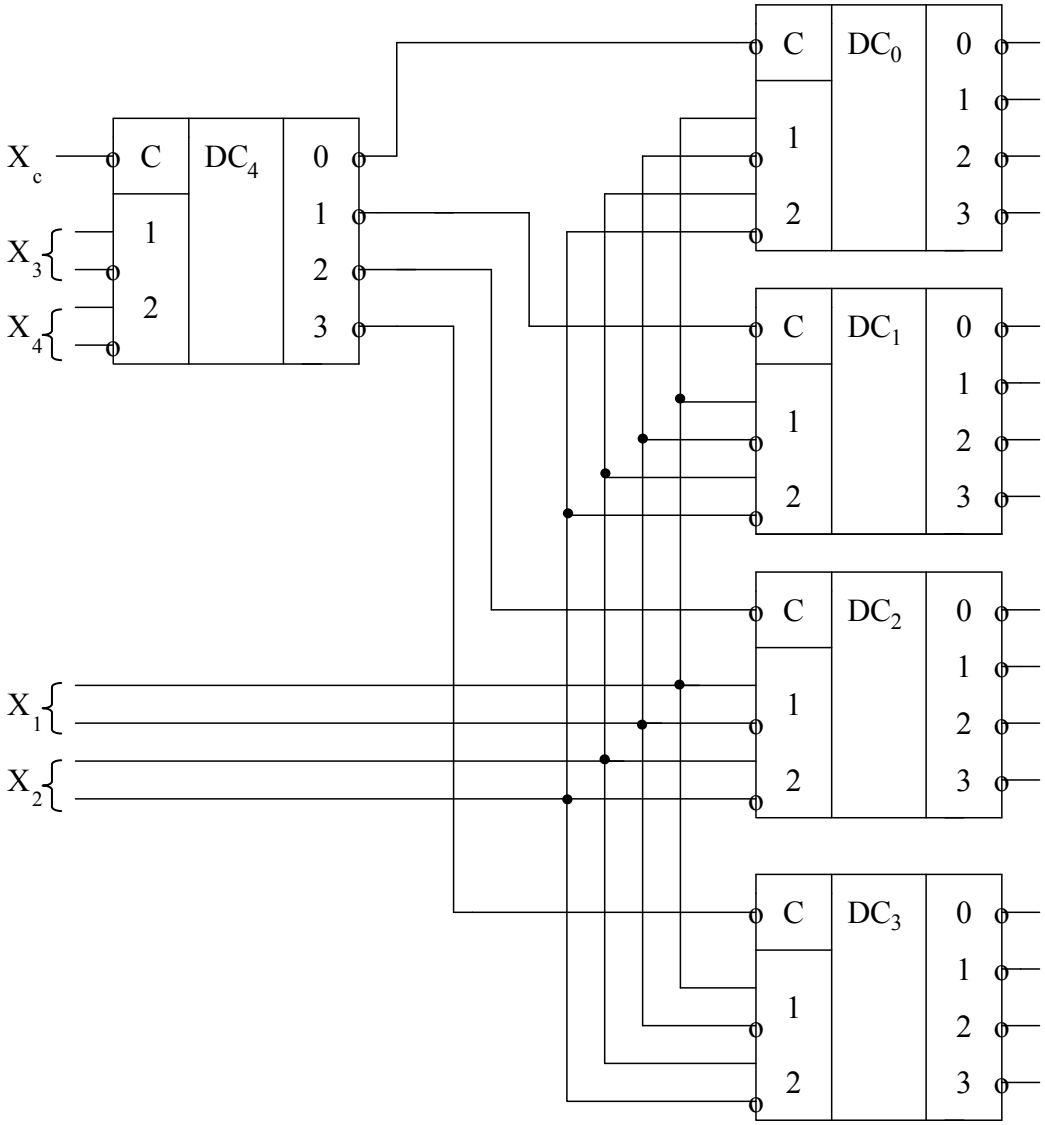


Рис. 3.9. Каскадное соединение дешифраторов

Таким образом, каскадно соединяя дешифраторы, можно строить схемы с большим числом выходов, используя в качестве составных частей малые дешифраторы.

3.5. Шифраторы

Шифраторы выполняют задачи обратные дешифраторам. С их помощью активным значением определяемых входных сигналов можно поставить в соответствие заданные выходные коды (комбинации выходных сигналов). Полный шифратор имеет n выходов и $m = 2^n$ входов (рис. 3.10, а). Выходной сигнал шифратора в числовой интерпретации соответствует номеру возбужденного входа. Таблица истинности одного

из вариантов восьмивходового полного шифратора с единичными активными значениями входных сигналов представлена в таблице 3.6.

Для построения схемы шифратора рассмотрим закономерности формирования двоичных переменных на его выходах. Выход y_1 соответствующий младшему разряду выходного кода, имеющему вес 1, должен принимать значение 1 при возбуждении любого из нечетных входов. Следовательно, это должен быть выход ЛЭ ИЛИ, к m/2 входам которого подключены все входы с нечетными номерами, то есть такими, двоичное представление номера которых в младшем разряде имеет 1 (рис. 3.10, б).

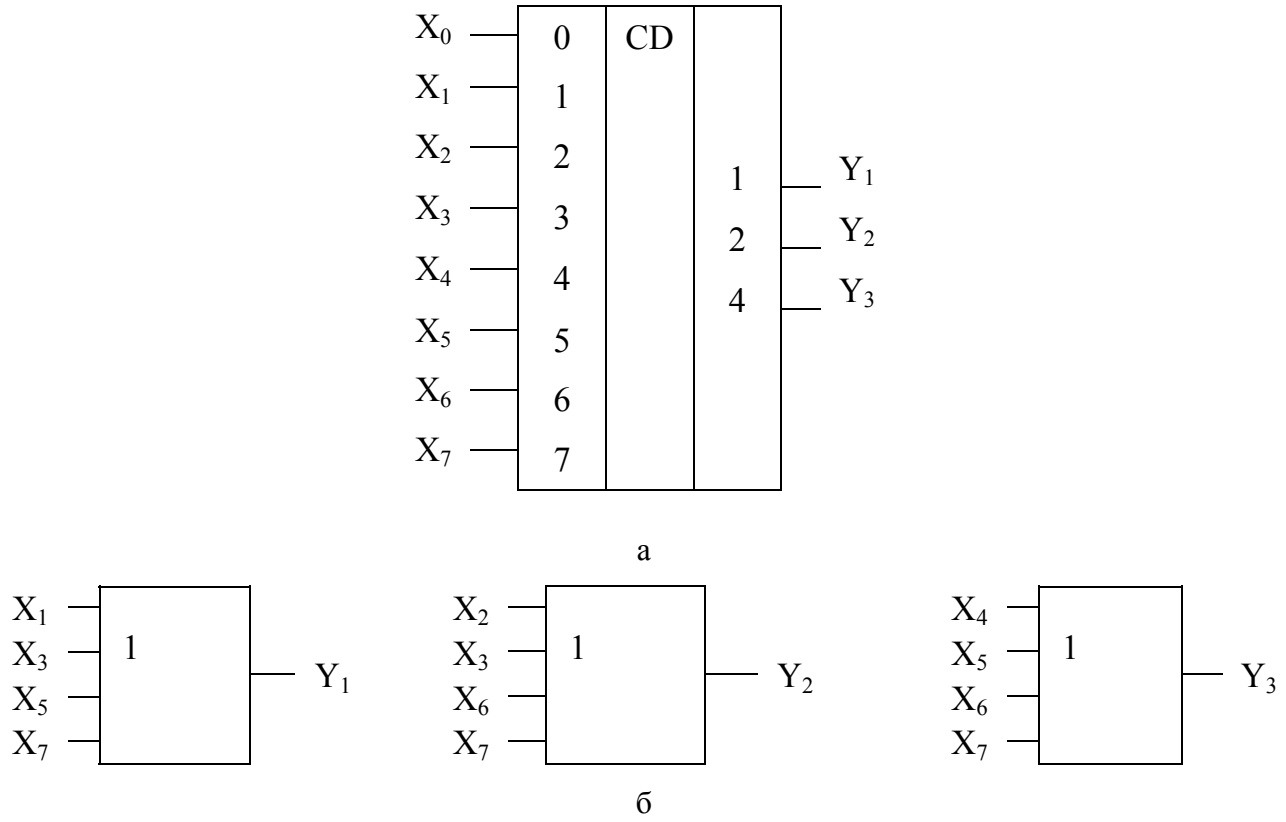


Рис. 3.10. Условное графическое обозначение шифратора (а);
построение шифратора с помощью схем ИЛИ (б)

Следующий выход Y_2, имеющий вес 2, должен возбуждаться при подаче сигнала на входы с номерами 2, 3, 6, 7, то есть с номерами, двоичное представление которых во втором по старшинству разряде имеет единицу. Следовательно, Y_2 также формируется элементом ИЛИ, имеющим m/2 входов. Таким образом, в общем случае Y_k формируется элементом ИЛИ с числом входов m/2, на который подаются те из входных переменных, двоичное представление номера которых в k-м разряде имеют единицу.

Таблица 3.6

X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	$У_3$	$У_2$	$У_1$
1	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0

При практической реализации шифратора может оказаться выгоднее использовать элементы И-НЕ. Тогда, в соответствии с двойственностью операций конъюнкции и дизъюнкции, входные переменные нужно брать с инверсией.

Рассмотренный шифратор обладает рядом недостатков:

1) невозможно наращивать информационную емкость (число входов и выходов), соединяя между собой шифраторы меньшей емкости;

2) для правильной работы шифратора необходимо, чтобы возбужден был всегда строго один из входов. Во всех других случаях получаемый на выходе сигнал не отражает реальной ситуации на входе.

От этих недостатков свободен приоритетный шифратор. Такой шифратор работает как и рассмотренный, когда возбужденным является единственный вход. Когда возбуждено несколько входов, на выходе формируется число, указывающее минимальный из номеров возбужденных входов. Так, если возбуждены входы 3, 5, 6, то на выходе установится код числа 3.

Чтобы выделить ситуацию, когда отсутствуют возбужденные входы, в приоритетном шифраторе имеется дополнительный выход P, называемый выходом признака невозбуждения. Двоичная переменная Y_p, формируемая на этом выходе, принимает значение 0, когда возбужден по крайней мере один из входов, и значение единицы в противном случае. Условное обозначение приоритетного шифратора с инверсными входами и выходами, имеющего m = 8 и n = 3, показано на рис. 3.11. Чтобы сделать возможным наращивание информационной емкости шифратора, в него введен дополнительный вход разрешения E.

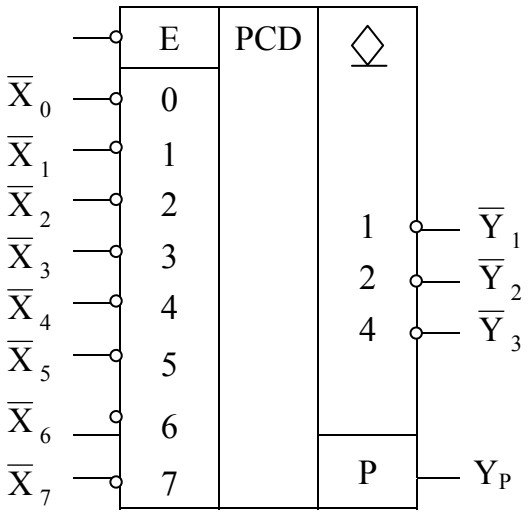


Рис. 3.11. Условное обозначение приоритетного шифратора

С той же целью выходы Y_1, Y_2, Y_3 выполняют либо с открытым коллектором, либо с тремя устойчивыми состояниями. Когда переменная X_E на разрешении равна 1, шифратор функционирует в соответствии с рассмотренным правилом. В противном случае сигнал на выходе P принимает значение 1, а все остальные выходы переводятся в непроводящее состояние.

3.6. Мультиплексоры

Мультиплексоры обеспечивают коммутацию на выходе одного из нескольких информационных входных сигналов в соответствии с заданным кодом на управляющих входах. Условное графическое обозначение одного из вариантов мультиплексора представлено на рис. 3.12. В зависимости от комбинации сигналов управления X_1, X_2, X_3 он обеспечивает коммутацию одного из восьми информационных входных сигналов $D_0 - D_7$ на выход Y.

Сигнал синхронизации V в данном случае имеет нулевое активное значение, разрешающее передачу информации с одного из входов на выход.

Структурная формула, определяющая функционирование рассматриваемого восьмивходового мультиплексора, имеет вид:

$$\begin{aligned} Y = {} & \overline{X}_3\overline{X}_2\overline{X}_1\overline{V}D_0 + \overline{X}_3\overline{X}_2X_1\overline{V}D_1 + \overline{X}_3X_2\overline{X}_1\overline{V}D_2 + \overline{X}_3X_2X_1\overline{V}D_3 + \\ & + X_3\overline{X}_2X_1\overline{V}D_4 + X_3\overline{X}_2\overline{X}_1\overline{V}D_5 + X_3X_2\overline{X}_1\overline{V}D_6 + X_3X_2X_1\overline{V}D_7. \end{aligned} \qquad (3.2)$$

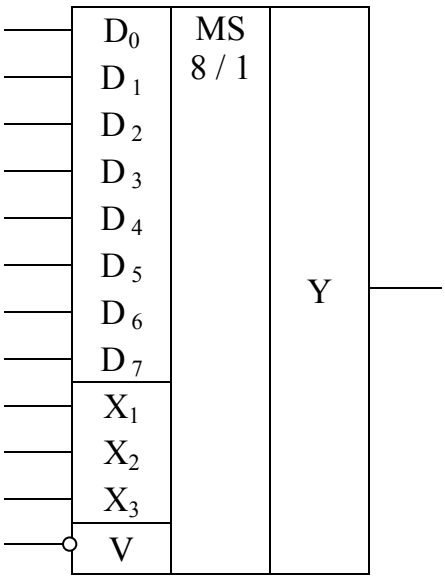


Рис. 3.12. Условное графическое обозначение мультиплексора

Мультиплексор можно реализовать с помощью ЛЭ заданного базиса. В его структуру можно ввести и более сложные цифровые устройства. Сопоставляя формулы (3.1) и (3.2), можно заметить, что для каждого входа D комбинации сигналов управления X_1, X_2, X_3 в мультиплексоре такие же, как и в дешифраторе. Следовательно, составной частью мультиплексора является дешифратор.

Структурная схема одного из возможных вариантов мультиплексора с четырьмя информационными входами $D_0 – D_3$ приведена на рис. 3.13.

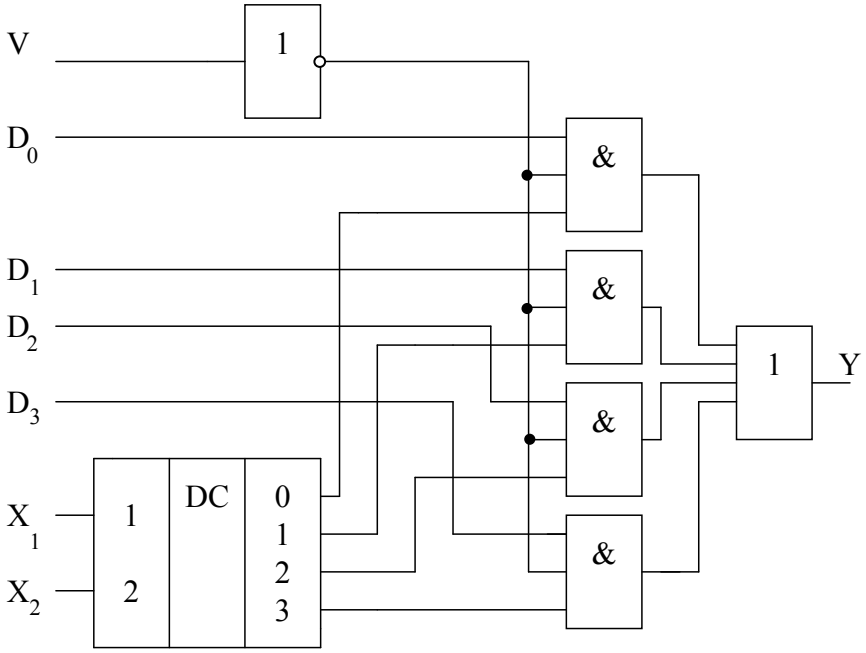


Рис. 3.13. Схема мультиплексора с четырьмя информационными входами

3.7. Демультиплексоры

Демультиплексоры представляют собой цифровые устройства для коммутации информационного входного сигнала в одном из нескольких выходов в соответствии с заданным кодом на управляющих входах. Иными словами, демультиплексоры решают задачи обратные мультиплексированию. Условное графическое обозначение демультиплексора с четырьмя информационными выходами представлено на рис. 3.14. Структурные формулы, определяющие формирование выходных сигналов такого демультиплексора, имеют вид:

$$Y_0 = \overline{X}_2\overline{X}_1 D;\ Y_1 = \overline{X}_2 X_1 D;\ Y_2 = X_2\overline{X}_1 D;\ Y_3 = X_2 X_1 D. \qquad (3.3)$$

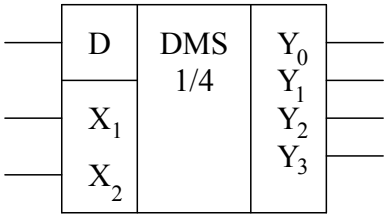


Рис. 3.14. Условное графическое обозначение демультиплексора

Точно такими же выражениями определяется формирование выходных сигналов дешифратора, использующего синхронизацию по входу D. Поэтому в качестве демультиплексора можно применять дешифратор, в котором информационный входной сигнал подан на вход синхронизации. Примером является реализация 2-разрядного демультиплексора с нулевыми активными сигналами на основе рассмотренного выше сдвоенного дешифратора К155ИД4 (см. рис. 3.6, б). Если первый разряд D_P подать на вход V_1, а второй разряд D_P – на вход V_3, то, в зависимости от комбинаций значений X_1, X_2, можно обеспечить коммутацию этого 2-разрядного кода $D_{P1}D_{P2}$ на любой из четырех выходов Y. На остальные входы синхронизации V_2 и V_4 можно подать постоянные значения ($V_2 = 1$, $V_4 = 0$), либо использовать их для организации синхронной передачи информации на выходы (рис. 3.15).

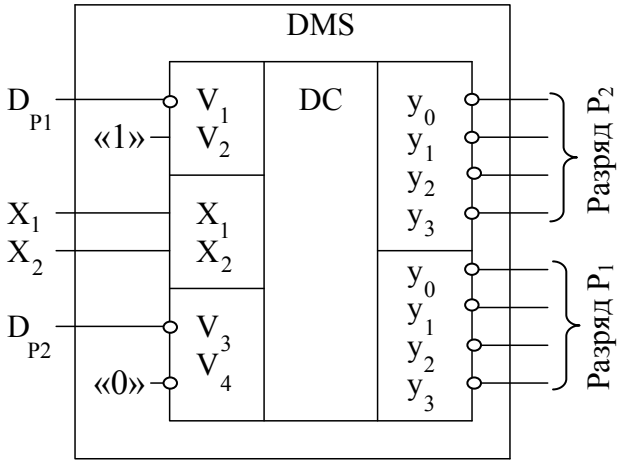
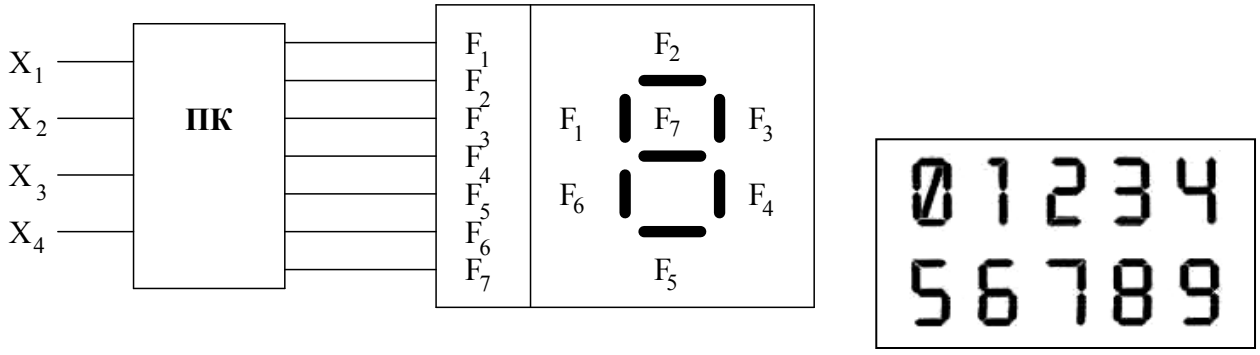


Рис. 3.15. Условное графическое обозначение демультиплексора на основе двух ИМС К155ИД4

3.8. Преобразователи кодов

В большинстве случаев задача перекодирования информации решается с помощью специальных КУ-преобразователей кодов. Синтез заданного преобразователя сводится к минимизации и реализации в выбранном базисе стольких функций, сколько разрядов содержится в выходном коде. Аргументами этих функций являются разряды входного кода. При наличии избыточных наборов их используют для доопределения и упрощения логической схемы.

Синтез преобразователя кодов рассмотрим на примере преобразования двоично-десятичного кода в специальный семиразрядный код, необходимый для управления работой семиэлементного цифрового индикатора, схема включения которого показана на рис. 3.16, а, а индикация цифр показана на рис. 3.16, б. Подсветка каждого из его элементов, образующих цифры от 0 до 9, производится подачей единичного сигнала на вход, соответствующий этому элементу. Сигналы поступают на цифровой индикатор с выхода преобразователя кодов (ПК). На входы преобразователя подаются логические сигналы четырехразрядных двоично-десятичных кодов. Задача заключается в синтезе такой логической схемы ПК, которая сформирует на индикаторе изображения цифр, соответствующих входным двоично-десятичным кодам.



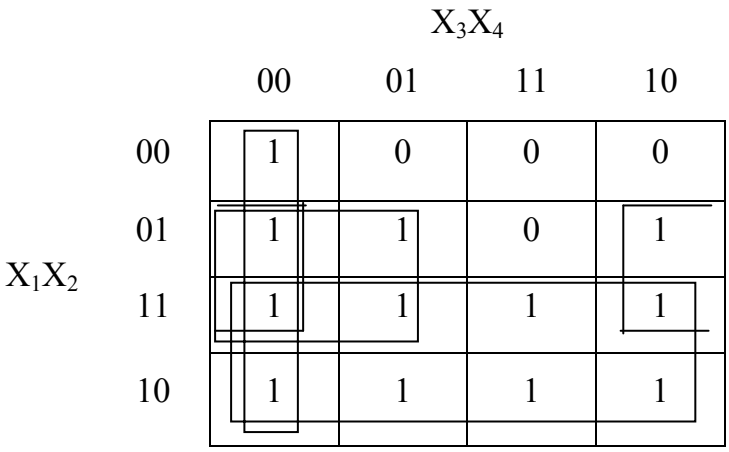
а б

Рис. 3.16. Структурная схема преобразователя кодов (а)
и индикация цифр на семисегментном индикаторе (б)

Выходные логические переменные F_1, ..., F_7 являются функциями входных переменных X_1, ..., X_4. Таблица истинности этих логических функций представлена в таблице 3.7.

Единичные значения выходных переменных соответствуют элементам, подсвечиваемым при изображении цифры. Например, десятичная цифра 0 задается двоично-десятичным кодом 0000. При этом подсвечиваются все элементы, за исключением F_7. Цифра 7 задаемся кодом 0111, а подсвечиваются элементы F_2, F_3 и F_4. Логические функции в таблице 3.7 определены не полностью: последние шесть кодовых комбинаций не соответствуют никаким десятичным цифрам и являются запретными.

Следуя изложенной выше методике, изобразим на рис. 3.17 карту Карно для логической функции F_1 .



$$F_1 = X_1 + X_2\overline{X}_4 + X_2\overline{X}_3 + \overline{X}_3\overline{X}_4$$

Рис. 3.17. Структура карты Карно для функции F_1

Таблица 3.7

Цифра	x_1	x_2	x_3	x_4	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	0	1	1	1
3	0	0	1	1	0	1	1	1	1	0	1
4	0	1	0	0	1	0	1	1	0	0	1
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	0	1	1	1	1
7	0	1	1	1	0	1	1	1	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	0	1
-	1	0	1	0	*	*	*	*	*	*	*
-	1	0	1	1	*	*	*	*	*	*	*
-	1	1	0	0	*	*	*	*	*	*	*
-	1	1	0	1	*	*	*	*	*	*	*
-	1	1	1	0	*	*	*	*	*	*	*
-	1	1	1	1	*	*	*	*	*	*	*

Ячейки, в которых функция не определена, отметим знаком * и доопределим их единицами. Покрытие содержит четыре прямоугольника. Следовательно, минимизированная ДНФ этой функции должна содержать четыре элементарных произведения, что и показано на этом же рисунке.

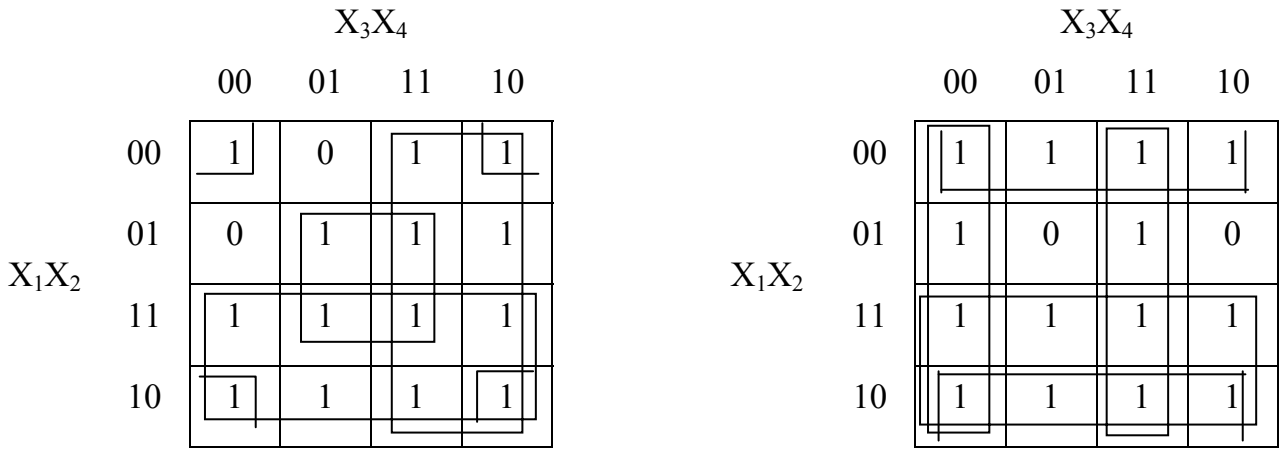
$$F_1 = X_1 + X_2\overline{X}_4 + X_2\overline{X}_3 + \overline{X}_3\overline{X}_4.$$

Аналогично, доопределив единицами все неопределенные значения функций F_2 и F_3, получим для них наилучшие покрытия и минимальные дизъюнктивные формы, показанные на рис. 3.18, а, б.

$$F_2 = X_1 + X_3 + X_2X_4 + \overline{X}_2\overline{X}_4;\ F_3 = X_1 + X_2 + X_3X_4 + \overline{X}_3\overline{X}_4.$$

При минимизации функции F_4, представленной на рис. 3.19, ячейку с координатами 1010 доопределим нулем. Формула минимизированной функции при этом существенно упростится.

$$F_4 = \overline{X}_3 + X_4 + X_2.$$



$F_2 = X_1 + X_3 + X_2X_4 + \overline{X}_2\overline{X}_4;$ $F_3 = X_1 + X_2 + X_3X_4 + \overline{X}_3\overline{X}_4$

а б

Рис. 3.18. Структура карты Карно для функции F_2 (а) и функции F_3 (б)

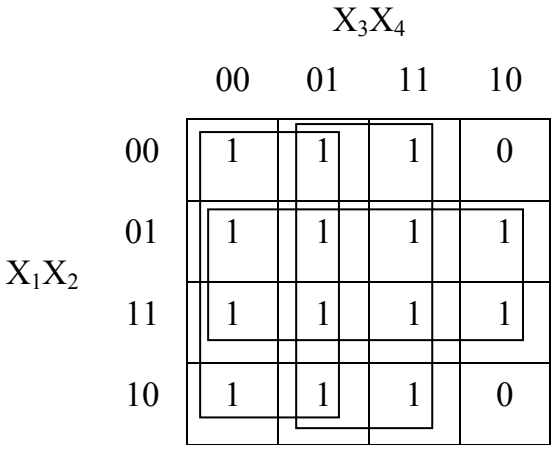


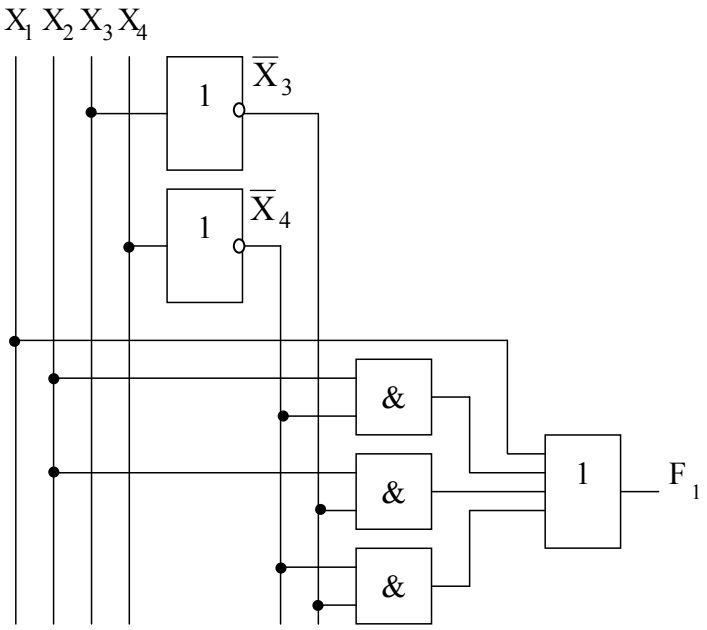
Рис. 3.19. Структура карты Карно для функции F_4

Каждая из рассмотренных функций может быть реализована в том или ином элементном базисе. Например, на рис. 3.20, а дана реализация функции F_1 в базисе элементов И, ИЛИ, НЕ, а на рис. 3.20, б — реализация этой же функции в базисе элементов И-НЕ.

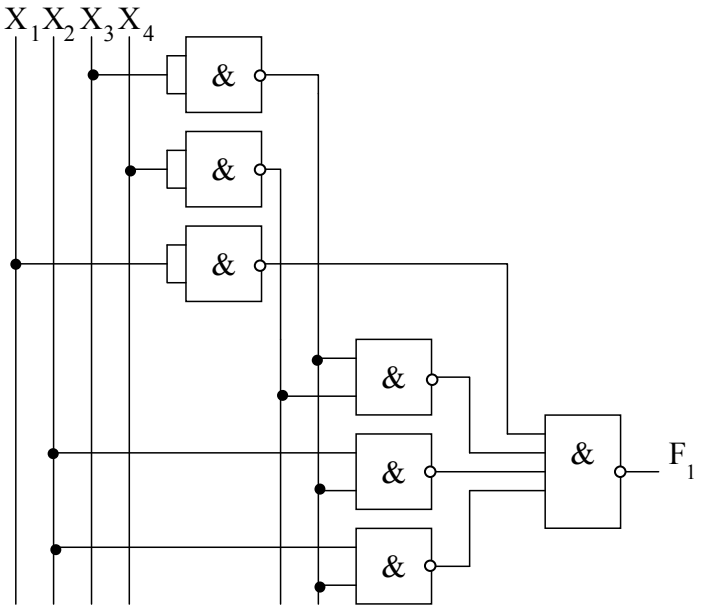
Соответствующая логическая формула легко устанавливается применением правила де Моргана:

$$F_1 = \overline{\overline{X_1 \vee X_2\overline{X}_4 \vee X_2\overline{X}_3 \vee \overline{X}_3\overline{X}_4}} = \overline{\overline{X}_1\overline{X_2\overline{X}_4}\,\overline{X_2\overline{X}_3}\,\overline{\overline{X}_3\overline{X}_4}}.$$

Аналогично реализуются и другие логические функции.



а



б

Рис. 3.20. Реализация функции F_1 в базисе И, ИЛИ, НЕ (а),
реализация функции F_1 в базисе И-НЕ (б)

Контрольные вопросы

1. Реализовать устройство неравнозначности в базисе И-НЕ.
2. Какие функции выполняет одноразрядный двоичный сумматор?
3. Какие существуют разновидности дешифраторов?
4. Какие функции выполняют мультиплексоры?
5. Найти выражения структурных формул для функций F_5, F_6, F_7 (п. 3.7).
6. Реализовать функции F_5, F_6, F_7 на ЛЭ в базисе И-НЕ.

4. ПОСЛЕДОВАТЕЛЬНОСТНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

Цифровое устройство называется последовательностным, если его выходные сигналы Y зависят не только от текущих значений входных сигналов X, но и от последовательности значений входных сигналов, поступивших на входы в предшествующие моменты времени. В последовательностных устройствах (ПУ) предыстория поступления последовательности входных сигналов обязательно фиксируется с помощью специальных запоминающих элементов или элементов памяти, поэтому говорят, что ПУ обладает памятью. Элементы памяти помимо входных и выходных сигналов характеризуются состоянием, которое может изменяться в дискретные моменты времени под воздействием сигналов на его входе. Простейший элемент памяти может принимать одно из двух состояний, например, 0 или 1. Это состояние может сохраняться длительный период времени. К цифровым последовательностным устройствам относят триггеры, регистры, счетчики. Эти устройства называются также цифровыми автоматами, конечными автоматами или автоматами с памятью.

4.1. Триггеры. Общее определение

Триггером называют устройство, которое может находиться неограниченно долго в одном из двух состояний устойчивого равновесия и переходить из одного состояния в другое под воздействием входного сигнала. Состояние триггера определяют по выходному сигналу. В нем может храниться либо 0, либо 1. Обычно триггер наряду с основным прямым выходом Q имеет еще инверсный выход P, потенциал которого имеет значение, обратное (в информационном смысле) значению прямого выхода, т. е. $P = \overline{Q}$ (если Q = l, то $P = \overline{Q} = 0$). Состоянию триггера 1 соответствует на выходе Q высокий уровень сигнала (1), а на выходе $\overline{Q}$ – низкий (0). Состоянию триггера 0 соответствует на выходе Q низкий уровень сигнала (0), а на выходе $\overline{Q}$ – высокий (1). Входы триггера подразделяются на информационные и вспомогательные (управляющие). Сигналы, поступающие на информационные входы, управляют состоянием триггера. Сигналы на вспомогательных входах используются для предварительной установки триггера в требуемое состояние и синхронизации.

Вспомогательные входы могут использоваться и в качестве информационных. Число входов триггера зависит от его структуры и назначения. Информационные входы триггера принято обозначать буквами S, R, J, K, D, T, а управляющие входы – C и V. Для триггера имеется стандартное обозначение (рис. 4.1). Здесь S и R являются информационными входами, Q и $\overline{Q}$ – выходами, T – обозначает триггер.

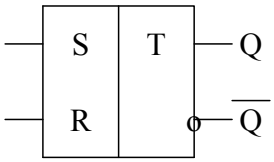


Рис. 4.1. Условное обозначение RS-триггера

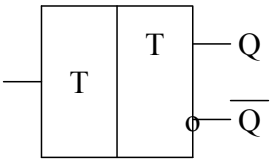


Рис. 4.2. Условное обозначение T-триггера

Триггер на рис. 4.1 называют RS-триггером или триггером с раздельным запуском. При раздельном запуске управляющие сигналы поступают на два входа триггера от двух источников сигнала. При поступлении управляющего сигнала на вход S (установка 1) триггер устанавливается в состояние 1 (т. е. Q = l, $\overline{Q}$ = 0), а при поступлении управляющего сигнала на вход R (установка 0) триггер устанавливается в состояние 0. На рис. 4.2 представлен триггер с общим (счетным) входом или T-триггер. При общем запуске управляющие сигналы поступают на один общий вход и при этом триггер изменяет состояние от каждого сигнала, т. е. каждый входной сигнал должен изменить состояние триггера на противоположное.

4.2. Классификация триггеров

Триггеры классифицируются по способу приема информации, по принципу построения, по функциональным возможностям. По способу приема информации триггеры подразделяются на асинхронные и синхронные. Асинхронные триггеры воспринимают информационные сигналы и реагируют на них в момент появления на входах триггера. Синхронные триггеры реагируют на информационные сигналы при наличии разрешающего сигнала на специальном управляющем входе C, называемом входом синхронизации. Синхронные триггеры подразделяются на триггеры со статическим и динамическим управлением по входу C. Триггеры со статическим управлением

воспринимают информационные сигналы при подаче на С-вход уровня 1 (прямой С-вход) или 0 (инверсный С-вход). Триггеры с динамическим управлением воспринимают информационные сигналы при изменении сигнала на С-входе от 0 к 1 (прямой динамический С-вход) или от 1 к 0 (инверсный динамический С-вход).

По принципу построения триггеры со статическим управлением можно разделить на одноступенчатые и двухступенчатые. Одноступенчатые триггеры характеризуются наличием одной ступени запоминания информации. В двухступенчатых триггерах имеется две ступени запоминания информации. Вначале информация записывается в первую ступень, а затем переписывается во вторую и появляется на выходе.

По функциональным возможностям различаются:

а) триггер с раздельной установкой состояний 0 и 1 (RS-триггер);

б) триггер с приемом информации по одному входу D (D-триггер или триггер задержки);

в) триггер со счетным входом Т (Т-триггер);

г) универсальный триггер с информационными входами J и К (JK-триггер).

Триггеры характеризуются быстродействием, чувствительностью, потребляемой мощностью, помехоустойчивостью, функциональными возможностями. Быстродействие определяется максимальной частотой переключения состояний триггера и достигает тысяч мегагерц. Чувствительность триггера определяется наименьшим напряжением на входе (пороговым напряжением), при котором происходит переключение. Помехоустойчивость характеризует способность триггера нормально работать в условиях помех. Функциональные возможности триггера характеризуются числом входных сигналов. Для обозначения функциональных возможностей триггеров в интегральном исполнении используется следующая маркировка: ТР – RS-триггер; ТВ – JK-триггер; ТМ – D-триггер. Для полного описания триггера достаточно задать его структурную схему из базовых логических элементов и закон функционирования. В качестве базовых логических элементов можно использовать элементы ИЛИ-НЕ, И-НЕ. Закон функционирования триггера может быть задан таблицей переходов, в которой входные сигналы в момент их изменения и состояние триггера обозначены индексом t, а после переключения – индексом $t + 1$.

4.3. Асинхронный RS-триггер с прямыми входами

Эти триггеры имеют два информационных входа R и S, используемые для установки соответственно 0 и 1, а также два выхода – прямой Q и инверсный $\overline{Q}$. RS-триггер может быть построен на двух логических элементах ИЛИ-НЕ, соединенных в контур (рис. 4.3, а). Графическое обозначение приведено на рис. 4.3, б, закон функционирования описан в таблице 4.1.

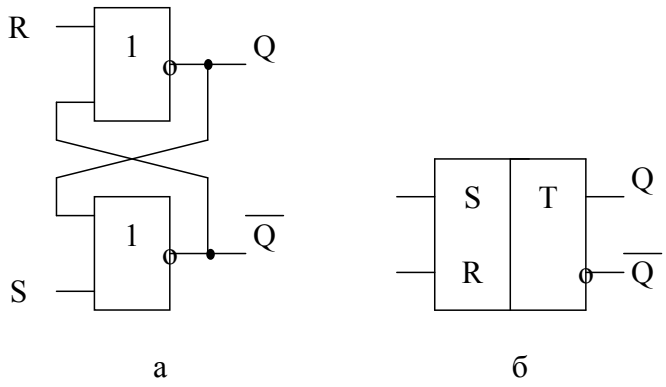


Рис. 4.3. Схема RS-триггера на элементах ИЛИ-НЕ (а)
и его условное графическое изображение (б)

Таблица 4.1

S^t	R^t	Q^t	Q^{t+1}	Режим
0	0	0	0	Хранение
0	0	1	1	
0	1	0	0	Установка 0
0	1	1	0	
1	0	0	1	Установка 1
1	0	1	1	
1	1	0	-	Запрещено
1	1	1	-	

Как следует из таблицы, при комбинации сигналов S = 1, R = 0 триггер переходит в состояние 1 ($Q^{t+1} = 1$) независимо от предыдущего состояния Q^t. При наборе сигналов $S^t = 0$, $R^t = 1$ триггер устанавливается в состояние 0 ($Q^{t+1} = 0$). Комбинация сигналов $S^t = 0$, $R^t = 0$ не изменяет состояния триггера. Набор сигналов $S^t = 1$, $R^t = 1$

является запрещенным, так как он приводит к нарушению работы триггера и неопределенности его состояния.

В RS-триггерах с прямыми входами управляющим воздействием обладают единичные уровни сигналов. Сигналы, которые приводят к переключению элемента, называют активными. Для элемента ИЛИ-НЕ, используемого для построения RS-триггера, активным сигналом является уровень 1.

Таблица состояний триггера в моменты t+1 может быть задана с помощью карты Карно (рис. 4.4). Используя карту Карно, можно построить минимальную булеву функцию для описания функционирования RS-триггера:

$$Q^{t+1} = S^t \vee Q^t \overline{R}^t.$$

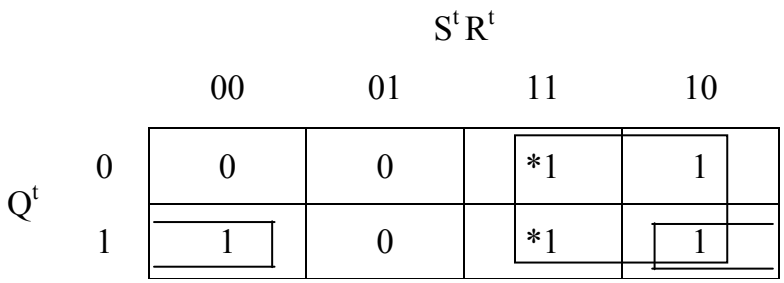


Рис. 4.4. Структура карты Карно

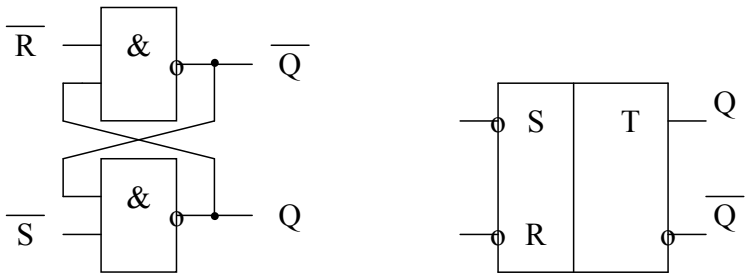
Из полученного выражения видно, что RS-триггер устанавливается в состояние 1 при воздействии входного уровня S = 1 либо остается в состоянии 1, если R = 0 и триггер был в состоянии Q = 1.

4.4. Асинхронный RS-триггер с инверсными входами

Триггеры такого типа строятся на логических элементах И-НЕ. В этом случае уровень 0 является активным входным сигналом, а уровень 1 – пассивным. Информационные входы и соответствующие сигналы таких триггеров принято обозначать как инверсные ($\overline{R}, \overline{S}$). Схема RS-триггера с инверсными входами приведена на рис. 4.5, а, условное графическое изображение триггера дано на рис. 4.5, б.

Закон функционирования RS-триггера на элементах И-НЕ описывается в таблице 4.2, из которой следует, что комбинация S = R = 0 является запрещенной,

а набор S = R = 1 – нейтральным. Следовательно, если нулевые сигналы на обоих триггерах на элементах ИЛИ-НЕ составляют нейтральную комбинацию, то для триггера на элементах И-НЕ они запрещены. Триггер переходит в состояние 1 при сигналах S = 0, R = 1, а при сигналах S = 1, R = 0 – переходит в состояние 0.



а б

Рис. 4.5. Схема RS-триггера на элементах И-НЕ (а)
и его условное графическое изображение (б)

Таблица 4.2

S^t	R^t	Q^t	Q^{t+1}	Режим
0	0	0	-	Запрещено
0	0	1	-	
0	1	0	1	Установка 1
0	1	1	1	
1	0	0	0	Установка 0
1	0	1	0	
1	1	0	0	Хранение
1	1	1	1	

Быстродействие асинхронного RS-триггера определяется задержкой установки его состояния t_T, равной сумме задержек передачи сигнала через цепочку логических элементов t в каждом. В данном случае $t_T = 2t_{3ср}$.

4.5. Синхронный RS-триггер

Синхронный RS-триггер отличается от асинхронного наличием С-входа, на который поступают синхронизирующие (тактовые) сигналы. Синхронный триггер состоит из асинхронного RS-триггера и комбинации цифрового устройства (рис. 4.6, а).

Как видно из рис. 4.6, а, синхронный триггер построен на элементах И-НЕ. Схема 1 представляет собой комбинационную схему с тремя входами S, C, R и двумя выходами. Схема 2 представляет собой асинхронный RS-триггер на элементах И-НЕ.

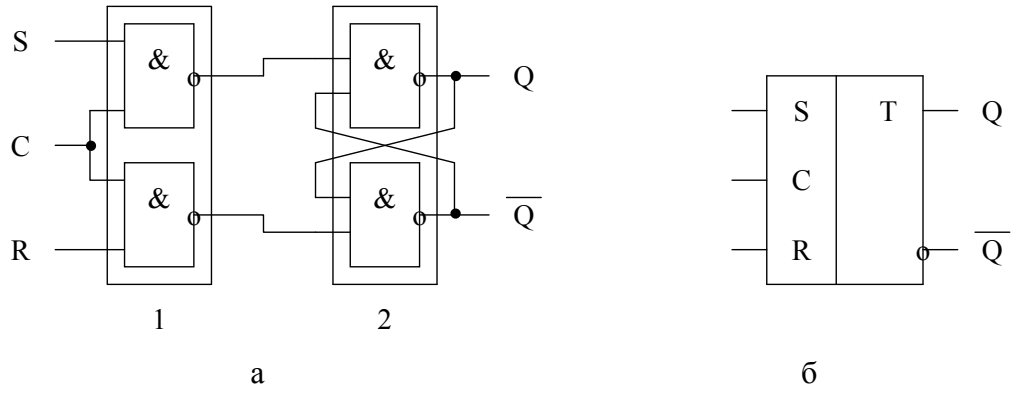


Рис. 4.6. Схема синхронного RS-триггера (а)
и его условное графическое обозначение (б)

При C = 0 выходы логических элементов схемы 1 принимают значение 1 и не зависят от входных сигналов S и R. При C = 1 входные логические схемы 1 открыты для передачи информационных сигналов R и S на входы асинхронного RS-триггера. Закон функционирования синхронного RS-триггера на элементах И-НЕ задан табл. 4.3.

Таблица 4.3

C^t	S^t	R^t	Q^t	Q^{t+1}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	-
1	1	1	1	-

Карта Карно, описывающая работу синхронного RS-триггера на элементах И-НЕ, представлена на рис. 4.7. Используя карту Карно, можно получить минимальную булеву функцию для описания работы синхронного RS-триггера:

$$Q^{t+1} = \overline{R}^t Q^t \vee \overline{C}^t Q^t \vee C^t S^t .$$

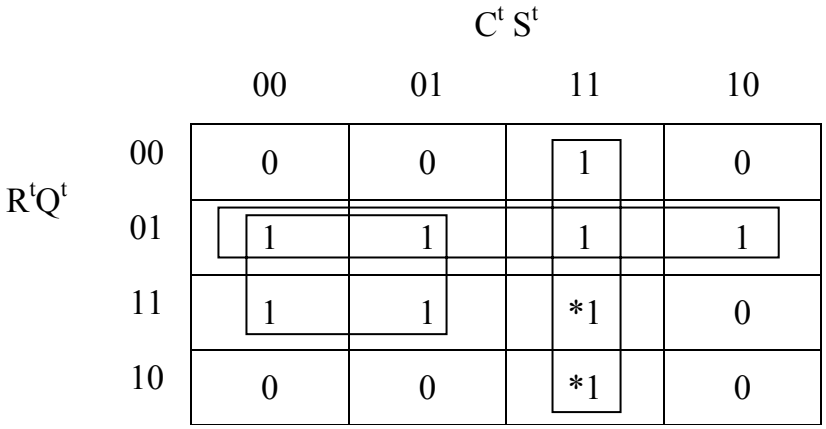


Рис. 4.7. Структура карты Карно для синхронного RS-триггера

Отсюда видно, что синхронный RS-триггер принимает состояние 1, если на входы C и S поступают уровни 1, или сохраняет единичное состояние Q при отсутствии единичных сигналов на входах C или R. Общее время установки состояния триггера t равно сумме задержек передачи сигнала через цепочку из трех логических элементов с задержкой t в каждом:

$$t = 3t_{3ср} .$$

При этом длительность синхросигнала t_c на входе C должна быть равна или превышать время переключения t_T.

Длительность паузы $t_п$ между двумя сигналами на входе C должна быть достаточной для переключения входных элементов в схеме 1 (рис. 4.6):

$$t_п \geq 3t_{3ср} .$$

Следовательно, минимальный период повторения синхронизирующих сигналов на входе C равен 4t, а наибольшая частота переключений

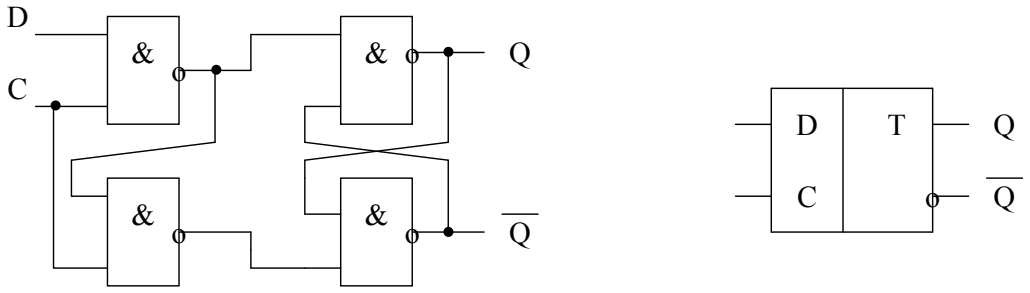
$$F_{max} = l/4t_{3ср} .$$

4.6. D-триггер

Триггер D-типа или триггер-задержка – это синхронный триггер с одним информационным входом D, реализующий логическую функцию

$$Q^{t+1} = C^t D^t + \overline{C}^t Q^t,$$

то есть значение сигнала на выходе Q триггера на t+1 такте (при Ct = 1) определяется значением входного сигнала D на предыдущем t такте. Основное назначение D-триггера заключается в задержке информации на один такт. Схема D-триггера на элементах И-НЕ приведена на рис. 4.8, а, а условное обозначение триггера – на рис. 4.8, б.

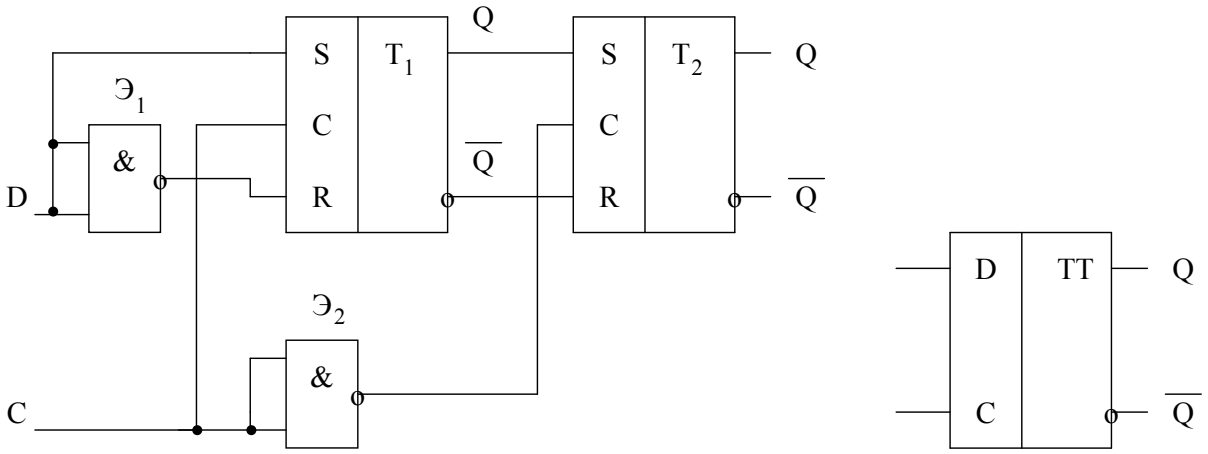


а б

Рис. 4.8. Схема D-триггера на элементах И-НЕ (а) и его условное графическое обозначение (б)

В момент времени t действия тактового импульса, соответствующего появлению сигнала 1 на входе D, на выходе триггера Q единичного напряжения еще нет: оно появится только после окончания тактового импульса и может быть использовано только при поступлении тактового импульса в момент времени t+1, т. е. с задержкой на один такт.

D-триггер (рис. 4.9, а) может быть построен на двух синхронных RS-триггерах T_1 и T_2 и двух инверторах $Э_1$ и $Э_2$. Такую схему триггера называют схемой «ведущий-ведомый» или двухступенчатым триггером. Его условное обозначение показано на рис. 4.9, б. Как и синхронные RS-триггеры T_1 и T_2, элементы $Э_1$ и $Э_2$ выполнены на элементах И-НЕ. Информационным входом триггера является вход D. Вход C служит для подачи тактовых импульсов. Пусть на вход D поступил сигнал 1. При D = 1 напряжение на входе T_1 соответствует сочетанию сигналов S = 1, R = 0. Появление очередного тактового импульса на входе C приведет к установлению триггера в состояние, при котором напряжение на его выходе равно единице.



а б

Рис. 4.9. Схема двухступенчатого триггера (а)
и его условное графическое обозначение (б)

На входах R и S триггера T_2 появляется сочетание сигналов S = l, R = 0. Однако во время действия тактового импульса напряжение на выходе инвертора $Э_2$ равно нулю. Сигнал на входе С триггера T_2 является нулевым и переключение триггера Т не происходит. Однако как только закончится тактовый импульс, сигнал на входе С триггера T_2 принимает значение 1 и триггер T_2 переключится в состояние единица. Если на входе D напряжение приняло нулевой уровень, то на выходе $Э_1$ напряжение соответствует единичному значению. На триггер T_1 подается сочетание входных сигналов S = 0, R = l, которое должно установить триггер T_1 в состояние 0. Это произойдет во время действия очередного тактового импульса, когда напряжение на выходе элемента $Э_2$ равно 0. Поэтому, хотя на входы триггера T_2 и будет во время действия тактового сигнала подаваться сочетание сигналов S = 0, R = 1, переключение триггера не произойдет. После окончания действия тактового импульса на выходе триггера T_2, напряжение примет нулевой уровень. Двухступенчатые D-триггеры обладают расширенными функциональными возможностями, например, при соединении инверсного выхода $\overline{Q}$ со входом D образуется триггер Т-типа.

4.7. Универсальный JK-триггер

Универсальный JK-триггер, схема которого приведена на рис. 4.10, представляет собой двухступенчатый синхронный триггер.

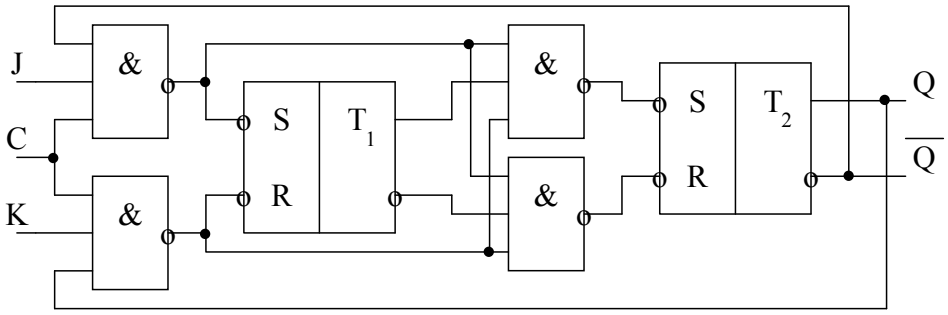


Рис. 4.10. Схема JK-триггера

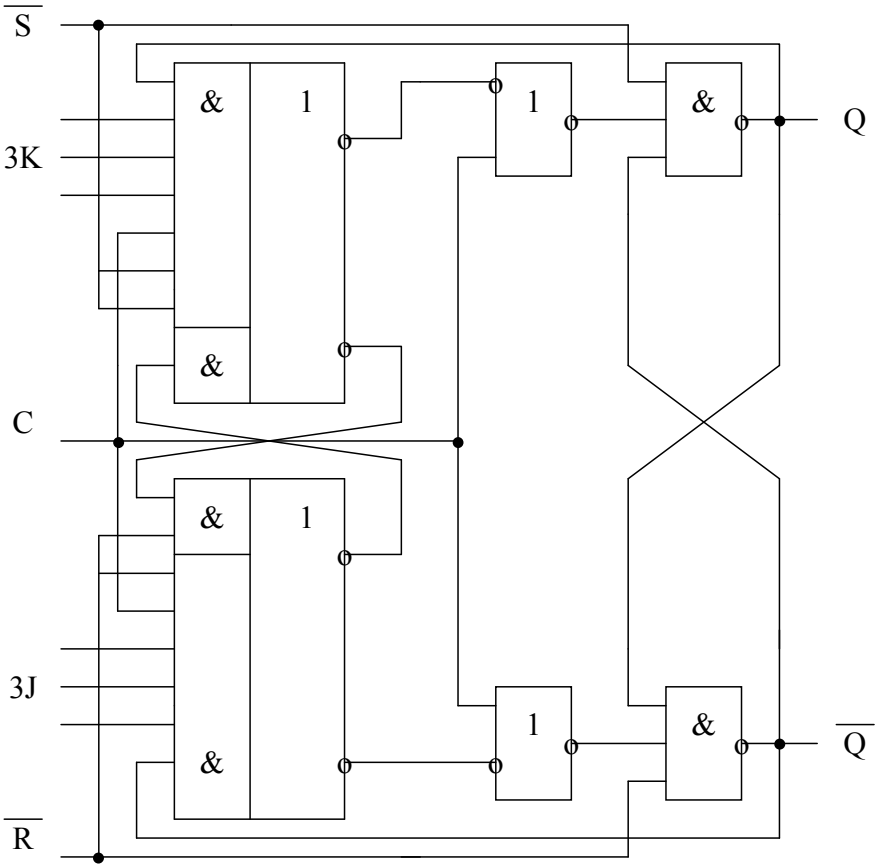
Схема JK-триггера состоит из двух асинхронных RS-триггеров с инверсными входами и двух комбинационных устройств, каждое из которых содержит две схемы И-НЕ с тремя входами каждая. Закон функционирования JK-триггера задается таблицей 4.4. JK-триггер отличается от синхронного RS-триггера тем, что не имеет запрещенных комбинаций сигналов на входах J и K. Кроме того, при J = K = 1 триггер изменяет свое состояние на противоположное, т. е. работает как триггер со счетным входом (T-триггер).

Таблица 4.4

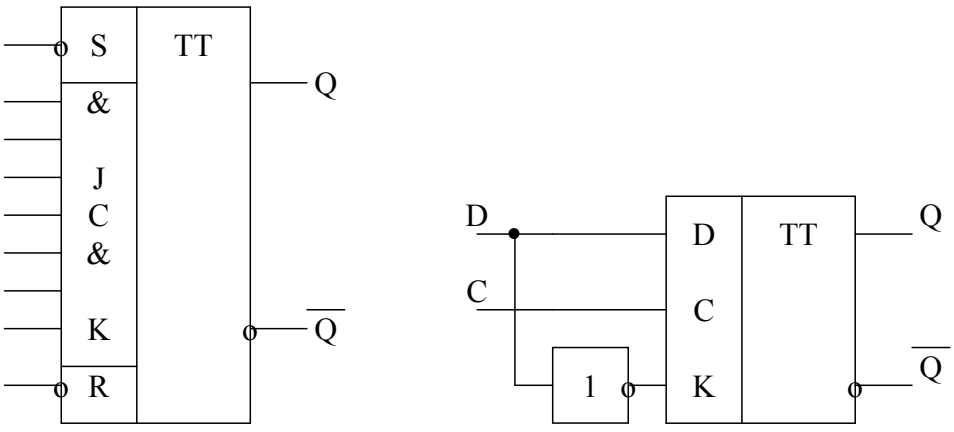
C^t	0	0	0	0	1	1	1	1
J^t	0	0	1	1	0	0	1	1
K^t	0	1	0	1	0	1	0	1
Q^t	Q^t	Q^t	Q^t	Q^t	Q^t	Q^t	Q^t	Q^t
Q^{t+1}	Q^t	Q^t	Q^t	Q^t	Q^t	0	1	$\overline{Q}_t$

При C = 0 входы J и K заблокированы и, следовательно, оказываются заблокированными входы S и R триггера T_1. При C = 1 в соответствии с информационными сигналами на входах J и K устанавливается состояние ведущего триггера T_1. При этом на входы S и R ведомого триггера T_2 поступают сигналы, при которых его предыдущее состояние сохраняется. При C = 0, когда входы триггера T_1 закрыты для входной

информации, входы триггера T_2 открываются и состояние ведущего триггера воспринимается ведомым триггером. Одним из широко используемых вариантов построения является схема JK-триггера с входной логикой (рис. 4.11, а). Условное обозначение триггера дано на рис. 4.11, б.



а



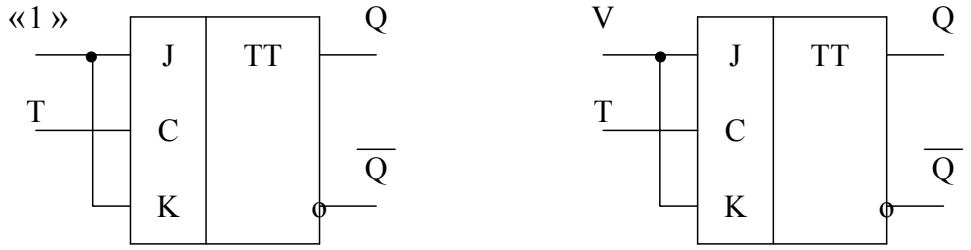
б в

Рис. 4.11. Схема JK-триггера со входной логикой (а);
его условное графическое обозначение (б); триггер D-типа на основе JK-триггера (в)

При C = 1 вторая ступень блокирована, а информационные сигналы устанавливают состояние первой ступени. При сигнале C = 0 вторая ступень воспринимает состояние первой. Требуемое начальное состояние триггера устанавливается с помощью сигналов, подаваемых на входы RS-триггеров первой и второй ступени. В схеме триггера имеются по три конъюнктивно связанных входа J и K. Их можно использовать для построения сложных схем. На основе JK-триггера можно построить триггер D-типа. Для этого информационный сигнал D подается на J-вход, а на вход K сигнал D подается через инвертор (рис. 4.11, в).

4.8. T-триггер

T-триггеры, или триггеры со счетным входом, могут быть получены из JK-триггера при соединении обоих информационных входов J и K, и подаче на них уровня 1 (рис. 4.12, а). Функционирование T-триггера описывается таблицей 4.5. В качестве счетного входа T используется вход C. При подаче сигнала на вход C T-триггер будет переключаться в состояние, противоположное предыдущему.



а б

Рис. 4.12. Схемы T-триггера (а) и TV-триггера (б) на основе JK-триггера

Таблица 4.5

C^t	J^t	K^t	Q^t	Q^{t+1}
0	1	1	Q^t	Q^t
1	1	1	Q^t	$\overline{Q}^t$

Разновидностью T-триггера является TV-триггер, в котором вход V является управляющим. При V = 1 TV-триггер превращается в T-триггер. При V = 0 TV-триггер сохраняет свое состояние неизменным (рис. 4.12, б).

4.9. Синхронный триггер с динамическим управлением

Синхронный триггер с динамическим управлением по входу С воспринимает информацию для изменения состояния лишь тогда, когда на С-входе совершается переход с уровня 0 на уровень 1, либо наоборот. Для получения RS-триггера с динамическим входом достаточно построить схему, показанную на рис. 4.13.

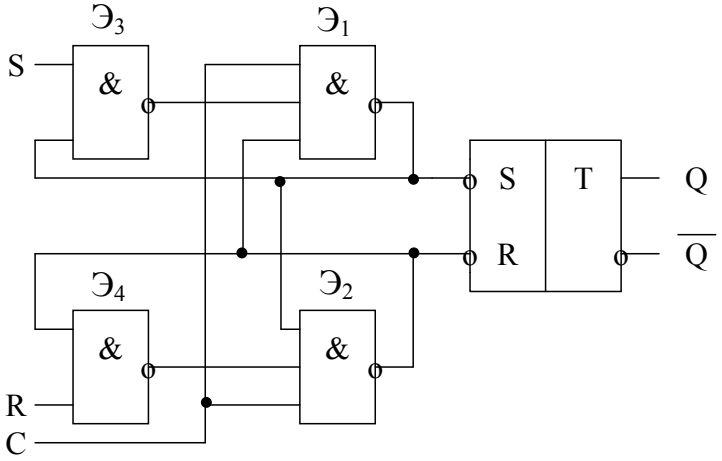
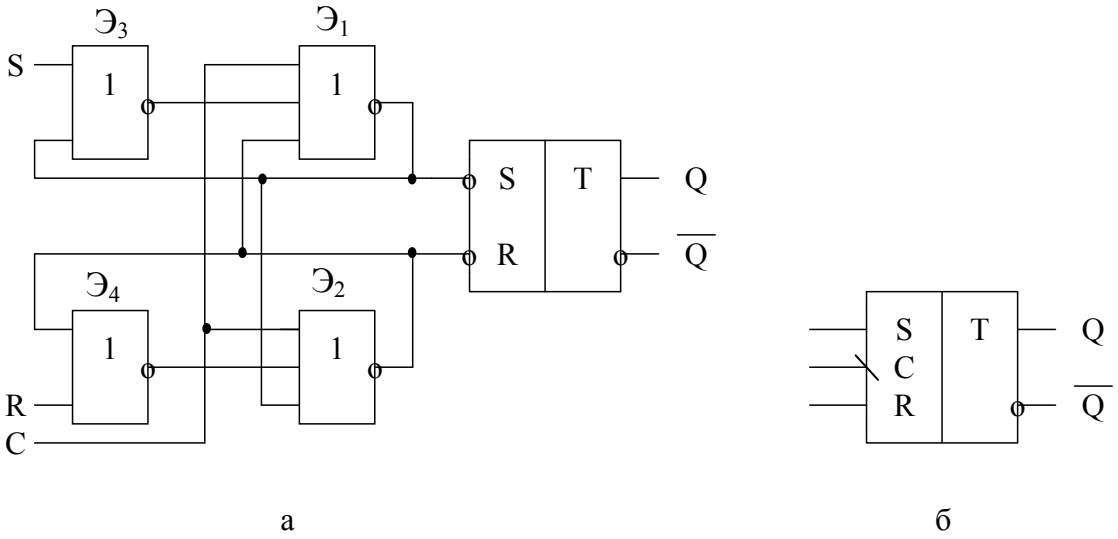


Рис. 4.13. Схема RS-триггера с динамическим входом на элементах И-НЕ

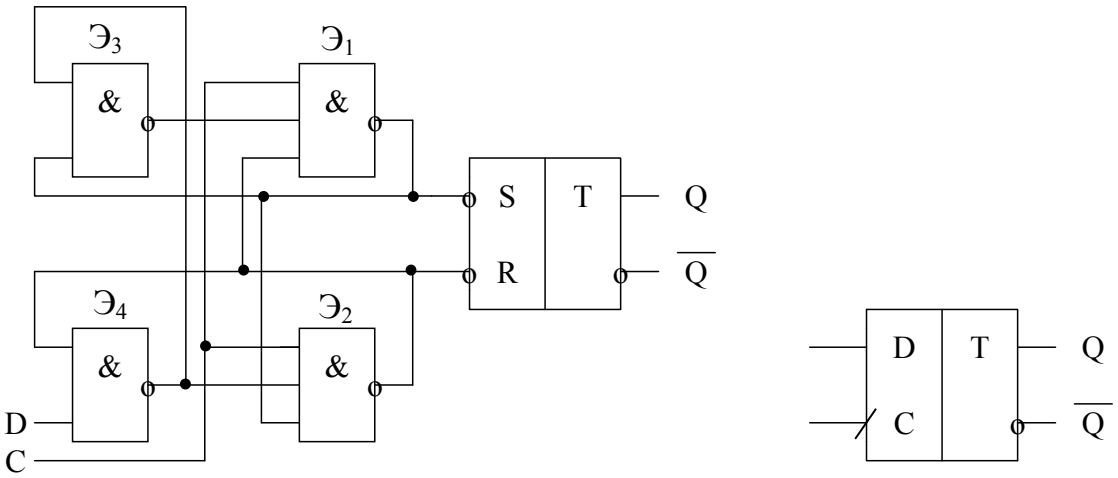
Если при С = 0 на информационные входы поступили какие-либо уровни S и R, то при смене уровня на входе С с 0 на 1 на выходе элемента $Э_1$ образуется 0, который поступает на вход элемента $Э_3$ и обеспечивает на его выходе уровень 1 независимо от последующих значений уровня на входе S. Вход S логически отключается и никакие изменения уровней на входах S и R триггер не воспринимает, пока не произойдет на входе С переход с уровня 0 на уровень 1. Аналогично можно построить схему RS-триггера с динамическим входом на элементах ИЛИ-НЕ (рис. 4.14, а). Здесь информация воспринимается триггером со входов S и R при смене уровней С = 1 на С = 0. Условное изображение такого триггера представлено на рис. 4.14, б.

Схема D-триггера с динамическим входом приведена на рис. 4.15, а, его условное обозначение дано на рис. 4.15, б. Прием в триггер информации со входа D происходит в момент смены на входе С уровня 0 на уровень 1.



а б

Рис. 4.14. Схема RS-триггера с динамическим входом на элементах ИЛИ-НЕ (а) и его условное графическое обозначение (б)



а б

Рис. 4.15. Схема D-триггера с динамическим входом на элементах И-НЕ (а) и его условное графическое обозначение (б)

4.10. Функции возбуждения триггеров

Таблицы истинности и функции переходов наиболее полно отражают все особенности работы триггеров. Каждая строка этих таблиц определяет, в какое состояние Q^{t+1} должен перейти триггер из состояния Q^t, если заданы сигналы на его управляющих входах. При проектировании устройств на триггерах часто решается обратная

задача: необходимо знать, какие логические сигналы следует подать на управляющие входы триггера, чтобы произошел его переход из одного заданного состояния (Q^t) в другое (Q^{t+1}). Для отображения такой информации часто используются таблицы функций возбуждения (табл. 4.6). Например: переход D-триггера из состояния $Q^t = 0$ в состояние $Q^{t+1} = 1$ происходит при единичном сигнале на его информационном входе D. Этот сигнал вызывает также переход триггера из состояния $Q^t = 1$ в состояние $Q^{t+1} = 1$. Аналогично составляется таблица функции возбуждения для Т-триггера.

Таблица 4.6

Переход	Функции возбуждения триггеров					
	D	T	RS		JK	
$Q^t \Rightarrow Q^{t+1}$	D	T	R	S	J	K
$0 \Rightarrow 0$	0	0	*	0	0	*
$0 \Rightarrow 1$	1	1	0	1	1	*
$1 \Rightarrow 0$	0	1	1	0	*	1
$1 \Rightarrow 1$	1	0	0	*	*	0

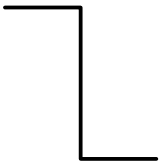
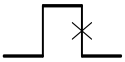
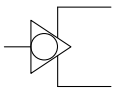
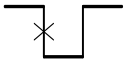
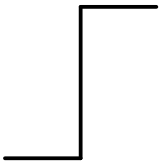
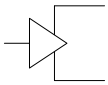
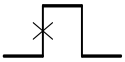
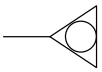
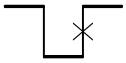
RS- и JK-триггеры содержат по два управляющих входа. Следовательно, каждый из них имеет по две функции возбуждения. Из таблицы истинности RS-триггера (табл. 4.3) следует, что триггер не изменяет своего нулевого состояния (переход 00) при двух различных значениях управляющего сигнала на входе R. Это означает, что сигнал на входе R не оказывает влияния на данный переход и значение сигнала на данном входе при переходе 00 не определено. Неопределенность функции возбуждения отмечается знаком *. Аналогичным знаком отмечена функция управляющего сигнала на входе S при переходе RS-триггера из состояния $Q^t = 1$ в состояние $Q^{t+1} = 1$. Функция возбуждения JK-триггера содержит по два неопределенных значения для каждого из управляющих сигналов на входах J и K.

Обозначения управляющих входов синхронных триггеров

Для полного цикла работы двухступенчатого триггера необходимы два перепада синхронизирующего сигнала. Для тактирования двухступенчатых синхронных триггеров могут использоваться импульсы как единичного активного уровня, так и

нулевого активного уровня. Символ ◁ на синхровходе триггеров означает, что перезапись информации и ведомый триггер происходит при переходе синхросигнала из единичного в нулевой логический уровень, а символ ▷ означает, что перезапись информации происходит при переходе синхросигнала из нулевого в единичное состояние. Для импульсов нулевого активного уровня внутри символов ставится кружок – знак инверсии (таблица 4.7).

Таблица 4.7

Спад – фронт импульса	Обозначение	Амплитуда тактового импульса	Уровень активности тактового импульса
		1	
		0	
		1	
		0	

4.11. Регистры

Регистром называется устройство, предназначенное для выполнения операций приема, хранения и передачи слов в двоичном коде. Наиболее распространены статические регистры. Каждому разряду слова, записанному в такой регистр соответствует свой разряд регистра, выполненный на основе статического триггера. Разряды регистра нумеруются в порядке нумерации разрядов в слове. Находят также применение динамические регистры, в которых функции запоминания значения двоичной переменной реализуются конденсатором (иногда индуктивностью). Ввиду того, что время хранения заряда на конденсаторе и тока в индуктивности ограничено, в динамических регистрах предусматривается операция регенерации (восстановления) хранимой информации. По способу записи информации регистры подразделяются на

параллельные, последовательные и параллельно-последовательные. В параллельном регистре запись и выдача слова осуществляется в параллельной форме одновременно во всех разрядах регистра. Последовательный регистр характеризуется последовательной записью и выдачей слова, начиная со старшего или младшего разряда. Параллельно-последовательный регистр имеет входы и выходы как для параллельной, так и для последовательной формы приема и передачи слова. На основе таких регистров осуществляются операции преобразования последовательного кода в параллельный и наоборот. В регистре могут выполняться следующие операции: прием слова из другого ПУ, передача слова из регистра в другие ПУ, поразрядные логические операции, сдвиг слова вправо или влево на заданное число разрядов, обращение кода.

4.11.1. Регистры памяти

Регистр с параллельным приемом и выдачей информации называется регистром памяти. Он позволяет записывать, хранить и в нужный момент выдавать информацию в прямом или обратном коде. Регистры памяти могут быть построены на RS-, D-, или JK-триггерах. Схема регистра памяти на RS-триггерах со счетным входом представлена на рис. 4.16.

При подаче управляющею импульса на шину «Сброс», все триггеры устанавливаются в нулевое состояние. Ввод новой информации в регистр осуществляется через ячейки И, связанные с входными шинами. Для записи информации, подведенной к входным шинам, подается управляющий импульс на шину «Ввод». При этом срабатывают те ячейки И, на входных шинах которых действует сигнал 1. Под действием импульсов, появляющихся на выходах ячеек И, соответствующие триггеры будут установлены в состояние 1. Вывод информации из регистра также осуществляется через элементы И, связанные с выходами триггеров.

Для вывода информации (считывания) управляющий сигнал подается на шину «Вывод». При этом срабатывают ячейки И, соединенные с триггерами, в которых записана 1 и информация в параллельном коде передается на выходные шины.

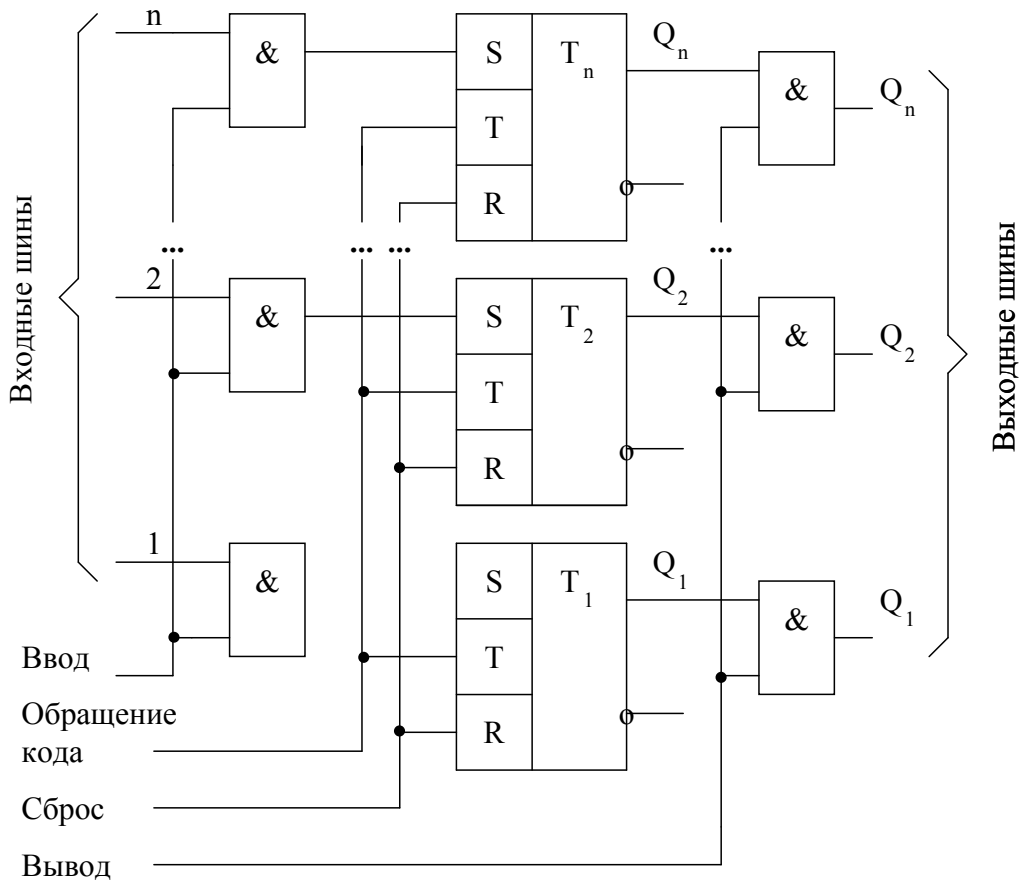


Рис. 4.16. Схема регистра памяти на RS-триггерах

Для выдачи информации в инвертированном (обратном) коде, когда все единицы заменяются нулями, а нули – единицами, необходимо подать управляющий импульс на шину «Обращение кода», соединенную со счетными входами триггеров. При этом состояние всех триггеров меняется на противоположное, и если теперь подать сигнал на шину «Вывод», то на выходные шины будет передана информация в обратном коде. Вторичное воздействие импульса «Обращение кода», если это необходимо, преобразует информацию снова в прямой код.

4.11.2. Сдвигающие регистры

Регистры с последовательной записью и выдачей информации называют сдвигающими регистрами. Сдвигающие регистры строятся на D-триггерах или JK-триггерах, включенных по схеме D-триггера. Они предназначены для сдвига слова, записанного в регистр, вправо или влево на один или несколько разрядов. Для

ввода информации в регистр, записываемое слово в виде последовательности импульсов, разряд за разрядом поступает на вход регистра в моменты действия тактовых импульсов. Для выдачи записанной информации необходимо снова подать продвигающие импульсы. При этом на выходе регистра, разряд за разрядом, начиная с младшего, в момент действия тактовых импульсов будут появляться сигналы, соответствующие кодовой комбинации, хранящейся в регистре. По мере вывода информации из регистра, старшие разряды регистра будут освобождаться. Поэтому вывод информации из регистра можно совместить с записью новой информации. Рассмотрим схему регистра на универсальных JK-триггерах (рис. 4.17).

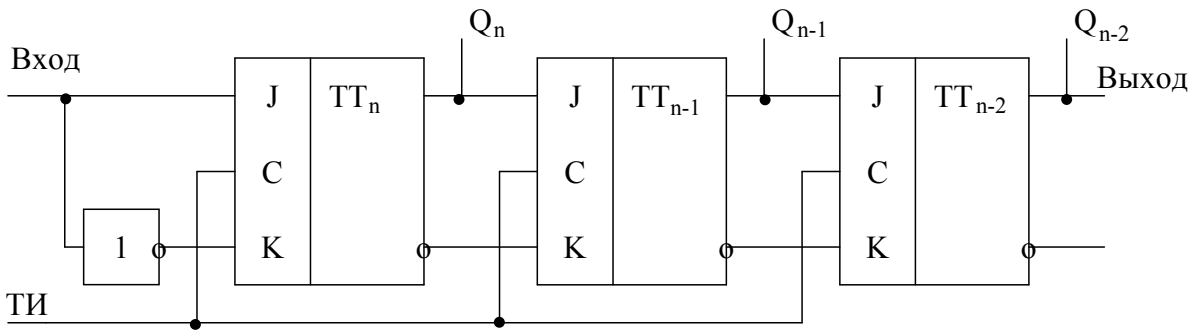


Рис. 4.17. Схема сдвигающего регистра на JK-триггерах

В момент поступления тактового импульса (ТИ) на синхронизирующие входы триггеров они принимают информацию от соседних слева триггеров, т. е. информация сдвигается на один разряд вправо. Если сдвиг информации возможен в обе стороны: и вправо, и влево – такие регистры называют реверсивными. Рассмотрим работу схемы реверсивного регистра на D-триггерах (рис. 4.18).

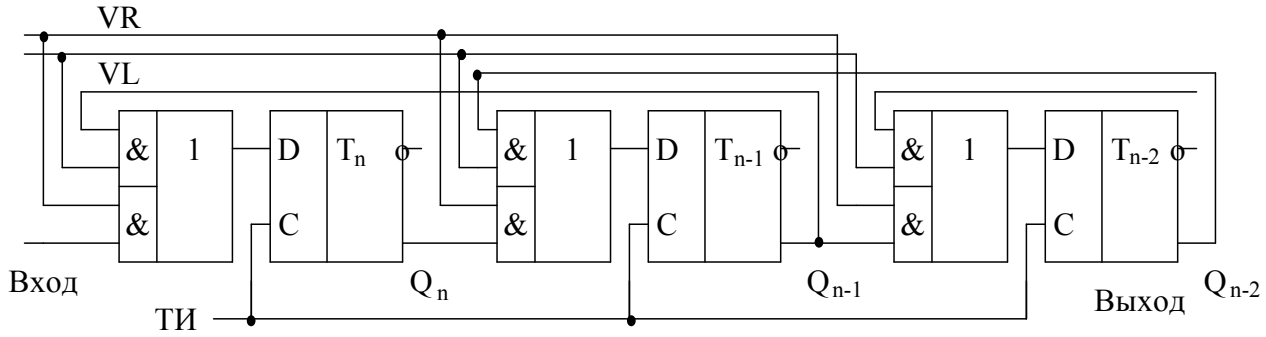


Рис. 4.18. Схема реверсивного регистра на D-триггерах

При подаче разрешающего сигнала на управляющий вход VR включается схема сдвига вправо. Реверсивный регистр при этом превращается в регистр сдвига вправо. При подаче разрешающего сигнала на управляющий вход VL включается схема сдвига влево. В регистрах сдвига влево и вправо разряды двоичного кода выходят за пределы разрядности регистра. Если соединить выход крайнего правого разряда регистра со входом крайнего левого разряда, то получим схему кольцевого регистра сдвига. Возможно также совмещение в одной схеме памяти и регистра сдвига. На рис. 4.19 представлена схема 4-разрядного регистра памяти и 4-разрядного регистра сдвига влево.

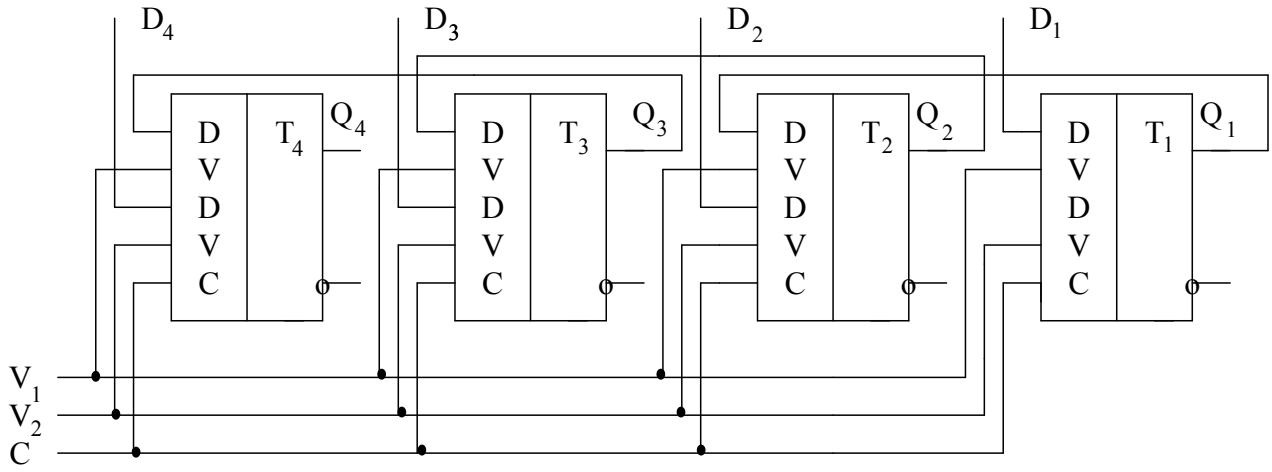


Рис. 4.19. Совмещенная схема регистра памяти и регистра сдвига

В каждом разряде регистра использован D-триггер с двумя D-входами и двумя управляющими V-входами. При подаче $V_1 = 0$, D-входы, используемые в схеме сдвига, отключаются, и схема превращается в схему регистра памяти с однофазными входами и парафазными выходами. При подаче $V_2 = 1$ разрешается прием информации, которая поступает на входы в параллельной форме и при поступлении синхроимпульсов на вход C записывается в регистр. Если поступает $V_1 = 1$, $V_2 = 1$, то схема превращается в схему регистра сдвига влево. В такой схеме информация принимается в параллельном виде, а выдача производится либо последовательно, либо параллельно.

4.12. Счетчики

Цифровым счетчиком называется устройство, осуществляющее счет числа входных импульсов и фиксирующие это число в каком-либо коде. После определенного числа импульсов счетчик обычно сбрасывается в исходное состояние и счет повторяется. Если на вход подается серия импульсов и выходной сигнал образуется только при появлении в счетчике одной заданной кодовой комбинации, то счетчик функционирует как делитель числа импульсов, т. к. эта кодовая комбинация периодически повторяется. В цифровой технике применяются суммирующие, вычитающие и реверсивные счетчики. По виду связи между разрядами различают счетчики с непосредственными связями, с параллельным переносом и комбинированными связями, асинхронные и синхронные. По коэффициенту счета счетчики разделяют на двоичные (бинарные) и с произвольным коэффициентом счета.

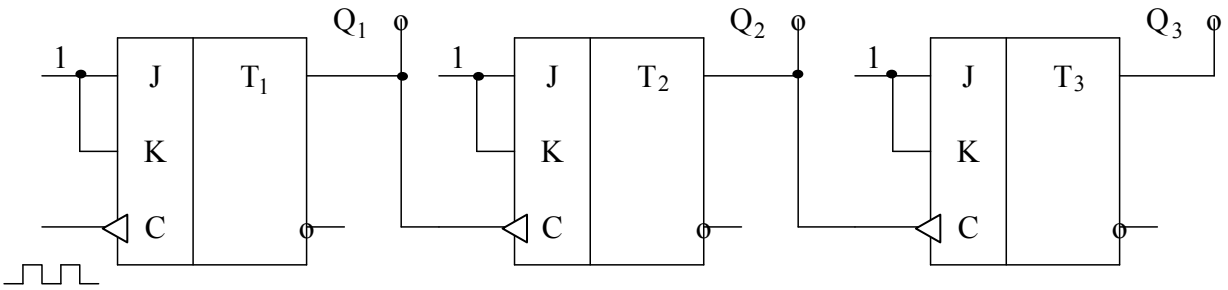
Основным элементом счетчиков является синхронный Т-триггер, используемый для работы в счетном режиме. Триггер делит на два частоту импульсов, поступающих на его вход (Т-вход для асинхронных, С-вход для синхронных триггеров). Каскадное включение n таких триггеров образует счетчик с коэффициентом пересчета 2^n, т. е. схему, которая при подаче 2^n импульсов возвращается к исходному состоянию. Такие схемы называют схемами счета по модулю 2.

На рис. 4.20, а изображен простейший способ включения триггеров (непосредственная связь), реализующий последовательный суммирующий счетчик, и временные диаграммы его работы (рис 4.20, б). Показан трехразрядный счетчик с коэффициентом пересчета $2^3 = 8$. Следовательно, после подачи на вход восьми импульсов счетчик возвратится к исходному состоянию.

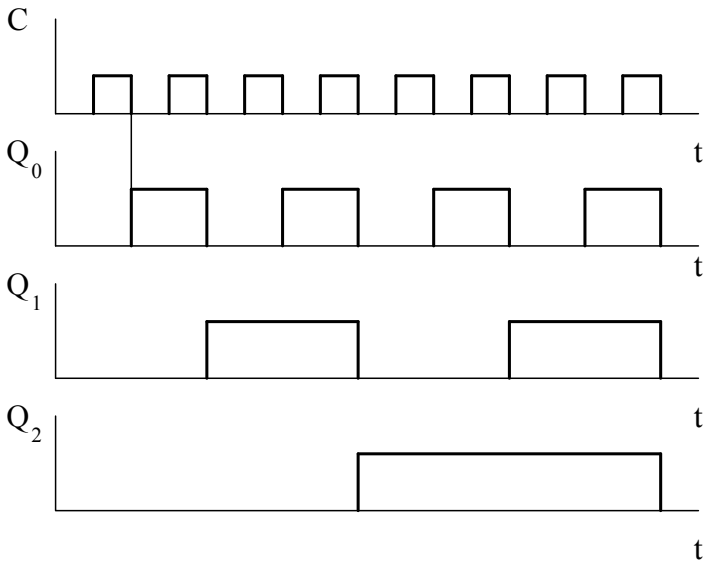
Возможен и другой вариант последовательного включения триггеров, когда их входы соединены с инверсными выходами предшествующих триггеров (рис. 4.21). Таким образом получают двоичный вычитающий счетчик.

Если в качестве исходного состояния счетчика выбрать код 111 (число 7), то последовательность входных импульсов уменьшает содержимое счетчика вплоть до 000, после чего наступает переключение, т. е. возврат к исходному состоянию 111.

Если в качестве исходного состояния счетчика принять код 000, то состояния входов триггера счетчика отображают отрицательное число сосчитанных импульсов, представленное в дополнительном коде.



а



б

Рис. 4.20. Схема суммирующего двоичного счетчика (а)
и временная диаграмма его работы (б)

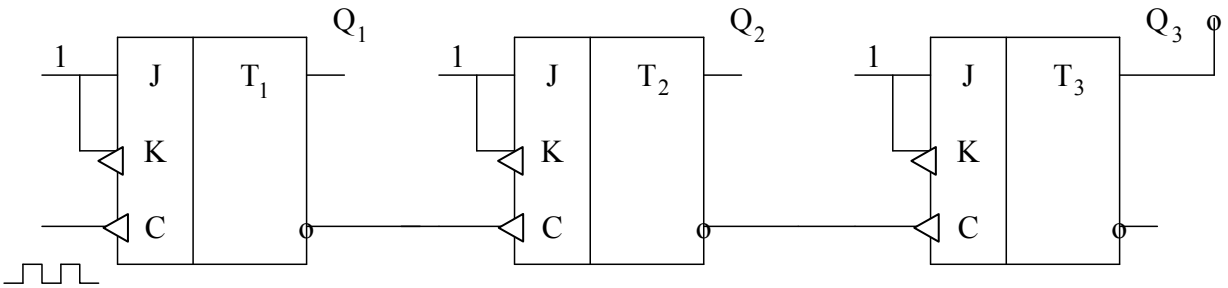


Рис. 4.21. Схема вычитающего двоичного счетчика

4.12.1. Параллельные счетчики

На рис. 4.20, а и 4.21 показаны схемы двоичных последовательных счетчиков, т. е. таких счетчиков, в которых при изменении состояния определенного триггера возбуждается последующий триггер, причем триггеры меняют свои состояния не одновременно, а последовательно. Если в данной ситуации должны изменить свои состояния n триггеров, то для завершения этого процесса потребуется n интервалов времени, соответствующих времени изменения состояния каждого из триггеров. Такой последовательный характер работы является причиной двух недостатков последовательного счетчика: меньшая скорость счета по сравнению с параллельными счетчиками и возможность появления ложных сигналов на выходе схемы. В параллельных счетчиках синхронизирующие сигналы поступают на все триггеры одновременно.

Последовательный характер переходов триггеров счетчика является источником ложных сигналов на его выходах. Например, в счетчике, ведущем счет в четырехразрядном двоичном коде с «весами» 8421, при переходе от числа $7_{10} = 0111_2$ к числу $8_{10} = 1000_2$ на выходе появится следующая последовательность сигналов:

$$0111 \Rightarrow 0110 \Rightarrow 0100 \Rightarrow 0000 \Rightarrow 1000.$$

Это означает, что при переходе из состояния 7 в состояние 8 на входах счетчика на короткое время появятся состояния 6; 4; 0. Эти дополнительные состояния могут вызвать ложную работу других устройств.

С целью уменьшения времени протекания переходных процессов, схему, приведенную на рис. 4.20, а, можно реализовать в варианте с подачей входных импульсов одновременно на все триггеры. В этом случае получим параллельный суммирующий счетчик (рис. 4.22).

Здесь на информационные входы триггеров подаются сигналы, являющиеся логической функцией состояния счетчика и определяющие конкретные триггеры, которые изменяют свое состояние при данном входном импульсе. Принцип стробирования сводится к следующему: триггер меняет свое состояние при поступлении очередного импульса синхронизации, если все предыдущие триггеры находились в состоянии логической единицы.

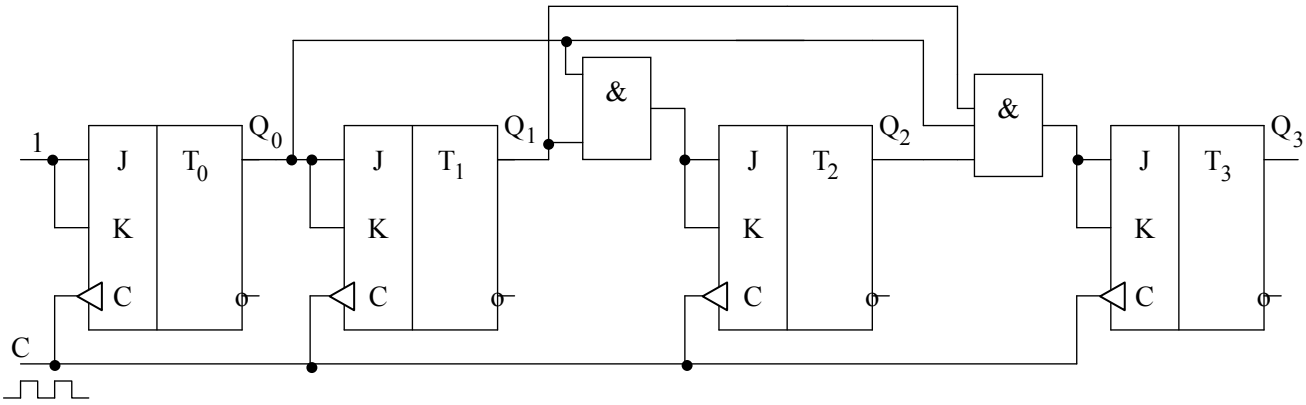


Рис. 4.22. Схема параллельного суммирующего счетчика

Параллельные счетчики имеют более высокое быстродействие по сравнению с последовательными, поскольку синхронизирующие импульсы поступают на все триггеры одновременно.

4.12.2. Реверсивные счетчики

Реверсивные счетчики позволяют суммировать или вычитать входные импульсы в зависимости от значений сигналов управления. Он реализуется путем объединения схем управления входами суммирующего и вычитающего счетчиков.

Реверсивные счетчики выпускаются в виде отдельных ИМС. Например, ИМС типа 155 ИЕ7 представляет собой реверсивный четырехразрядный двоичный счетчик. На рис. 4.23 представлена схема реверсивного восьмиразрядного двоичного счетчика, построенного на двух ИМС типа 155 ИЕ7. Разряд Q_8 является знаковым. Счет ведется в дополнительных кодах.

Каждая ИМС имеет по два счетных входа C_1 и C_2, обеспечивающих соответственно работу в режиме суммирования и вычитания. Изменения состояния счетчика происходит в момент поступления переднего фронта импульса на вход C_1 или C_2. Если счетчик работает в режиме суммирования, то изменяется сигнал на входе C_1. Сигнал на входе C_2 должен иметь при этом высокий логический уровень. При работе в режиме вычитания изменяется сигнал на входе C_2, а вход C_1 находится в состоянии логической единицы.

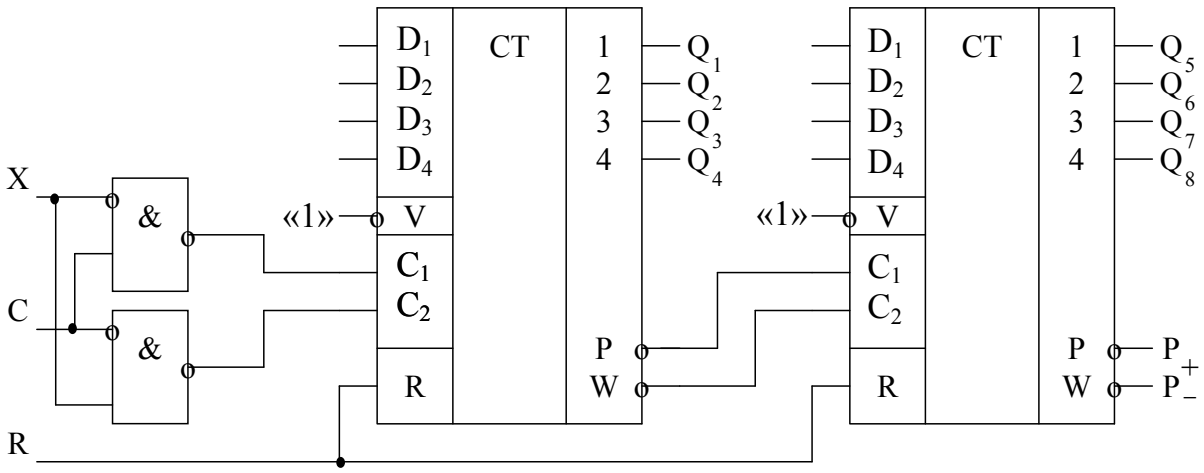


Рис. 4.23. Схема реверсивного восьмиразрядного счетчика на микросхемах 155ИЕ7

ИМС 155ИЕ7 имеет два выхода переполнения: выход P (переноса при суммировании) и выход W (заема при вычитании). Сигналы P и W формируются аналогичным образом. На выходе переноса P отрицательный спад импульса образуется при переполнении счетчика, т. е. при появлении па выходах $Q_1 – Q_4$ максимального числа и условии, что тактирующий сигнал на входе C имеет нулевое состояние. На выходе заема W импульс формируется при появлении на выходах всех разрядов $Q_1 – Q_4$ счетчика нулевого логического уровня, при этом вход C_2 должен находиться в состоянии логического нуля. Микросхема имеет входы $D_1 – D_4$, по которым возможна параллельная запись четырехразрядных кодов при условии, что на входы V и R поданы сигналы нулевого логического уровня.

Контрольные вопросы

1. Какое принципиальное отличие последовательностных устройств от комбинационных?

2. В чем отличие асинхронных и синхронных триггеров?

3. Какие входы может иметь регистр памяти?

4. Какие функции может выполнять регистр сдвига?

5. Какие способы используются для построения счетчиков по модулю не кратному степени два?

6. Какие основные недостатки счетчиков последовательного типа?

5. ЦИФРО-АНАЛОГОВЫЕ И АНАЛОГОВО-ЦИФРОВЫЕ ПРЕОБРАЗОВАТЕЛИ

5.1. Цифро-аналоговые преобразователи

Цифро-аналоговые преобразователи (ЦАП) – это преобразователи код-напряжение (ПКН). Задача рассматриваемого преобразователя состоит в выработке напряжения U на выходе ПКН, пропорционального входному двоичному числу N. Используя известную форму представления двоичных чисел

$$N = a_0 2^0 + a_1 2^1 + \ldots + a_{n-1} 2^{n-1} = \sum_{i=0}^{n-1} a_i 2^i ,$$

запишем операцию, выполняемую ПКН, в виде

$$U(N) = \frac{U_{max}}{N_{max}} N = U_{max} \sum_{i=0}^{n-1} \frac{2^i}{2^{n-1}} a_i = \sum_{i=0}^{n-1} U_i a_i , \qquad (5.1)$$

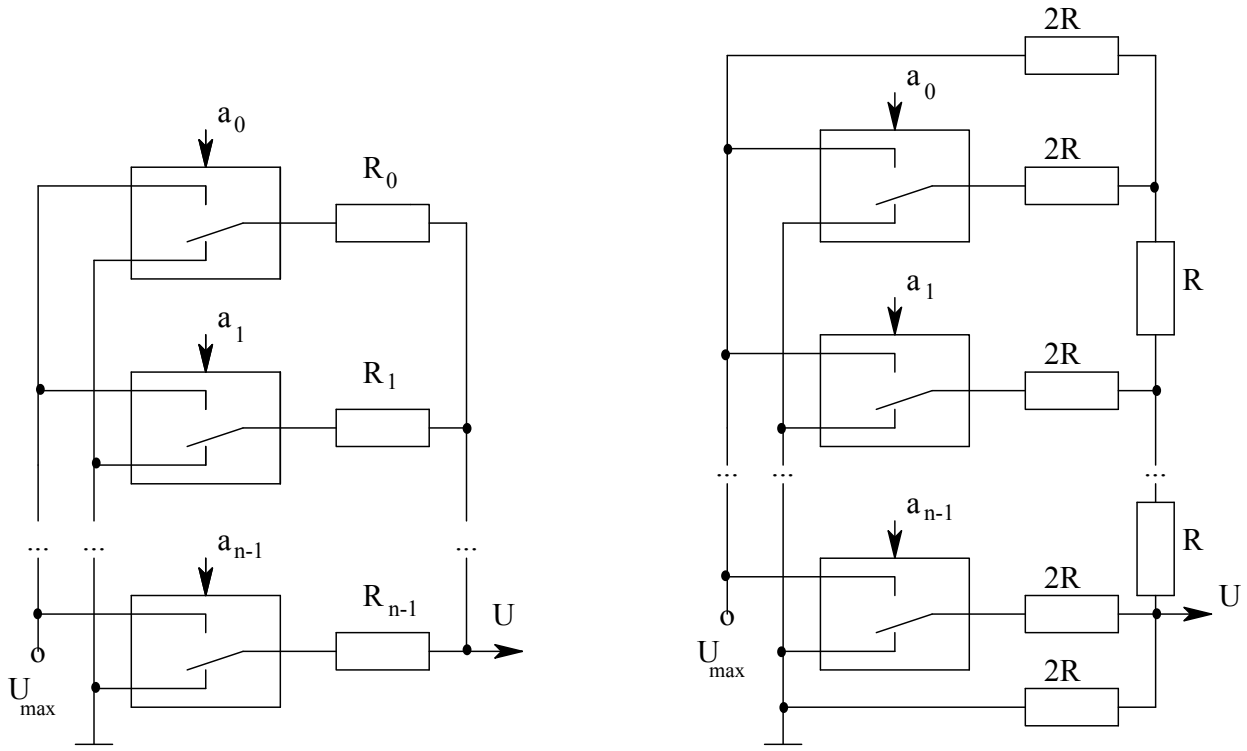
где отношение максимального выходного напряжения U_{max} к максимальному входному числу $N_{max} = 2^n - 1$ играет роль масштабного коэффициента, определяемого как напряжение, соответствующее единице входного числа, т. е. $U(1) = U_{max} / N_{max}$.

Из выражения (5.1) следует, что операция преобразования код-напряжение сводится к суммированию элементарных напряжений:

$$U_i = U_{max} 2^i / 2^n - 1 \approx U_{max} / 2^{n-i} = U_{max} K_i , \qquad (5.2)$$

которые образуются путем деления некоторого эталонного напряжения U_{max} с помощью резистивных делителей с коэффициентом передачи $K_i = 1/2^{n-i}$. При этом из операции суммирования исключаются те слагаемые U_i, которые соответствуют нулевым значениям элементов a_i, составляющих входное двоичное число ($a_{n-1}\ a_{n-2} \ldots a_1\ a_0$).

Для реализации этого принципа преобразования на практике используются две разновидности схем, представленные на рис. 5.1 а, б. Первую схему (рис. 5.1, а) называют ПКН с двоично взвешенными резистивными цепями (или схемой с суммированием напряжений), а вторую (рис. 5.1, б) ПКН с многозвенной резистивной цепью типа R-2R. В обеих схемах при появлении единицы в i-м разряде двоичного числа эталонное напряжение U_{max} проходит через соответствующий замкнутый ключ и резистивную цепь на выход. Различие этих схем в том, как формируется требуемый коэффициент деления эталонного напряжения с помощью резистивной матрицы.



а б

Рис. 5.1. Схемы ПКН с двоично взвешенными резистивными цепями (а) и многозвенной резистивной цепью типа R-2R (б)

В первой схеме (рис. 5.1, а) при $a_i = 1$ коэффициент передачи напряжения $K_i = U_i / U_{max}$ определяется отношением проводимости $Y_i = I / R_i$ к суммарной проводимости всей цепи $Y_\Sigma = \sum_{i=0}^{n-1} Y_i$. Значения сопротивлений R_i в этой схеме задаются следующим образом: $R_{n-1} = R$, $R_{n-2} = 2R$, $R_i = 2^{n-i-1}R$, ..., $R_0 = 2^{n-1}R$.

Тогда

$$Y_\Sigma = \frac{1}{R}\sum_{i=0}^{n-1} 2^{-(n-i-1)} \approx \frac{2}{R}$$

и, следовательно, $K_i = U_i / U_{max} = Y_i / Y = 1/2^{n-i}$. Это значит, что заданная совокупность сопротивлений удовлетворяет соотношению (5.2), которое определяет правило формирования коэффициентов передачи эталонного напряжения на выход ПКН.

По второй схеме ПКН (рис. 5.1, б) необходимый коэффициент передачи эталонного напряжения образуется с помощью многозвенного делителя напряжения.

В этой схеме коэффициент передачи напряжения от любого звена к следующему равен 1/2, за исключением последнего звена, где коэффициент равен 1/3.

Результирующий коэффициент передачи от i-го звена на выход

$$K_i = U_i / U_{max} = \frac{1}{3}\left(\frac{1}{2}\right)^{n-i-1} = \frac{2}{3}\left(\frac{1}{2}\right)^{n-i}. \qquad (5.3)$$

Таким образом, схема (рис. 5.1, б) также обеспечивает требуемое правило формирования коэффициентов, определяемое соотношением (5.3). Такая схема имеет коэффициент использования эталонного напряжения 2/3, в отличие от схемы (рис. 5.13, а), в которой этот коэффициент равен 1.

Однако, несмотря на этот недостаток и на большое число элементов схемы, преимуществом схемы ПКН с резистивной цепью типа R-2R является то, что для выполнения такой схемы требуется всего два номинала резисторов. Это существенно упрощает практическую реализацию таких схем, особенно при большой разрядности схем ПКН, когда предъявляются высокие требования к точности подбора номиналов резисторов. Поэтому схемы ПКН, использующие резистивные цепи типа R-2R, широко применяются на практике.

Одной из важнейших задач, связанных с разработкой и применением ПКН, является обеспечение требуемой скорости преобразования, или быстродействия схемы. Это важно тогда, когда ПКН является элементом более сложных устройств, например, преобразователей напряжения в код, характеристики которых в значительной степени зависят от характеристик используемых в них ПКН.

В рассмотренных схемах ПКН время выполнения операции преобразования определяется быстродействием ключевых схем и переходными процессами в резистивных цепях, обусловленными наличием паразитных емкостей. Второй фактор для этих схем является основным, так как значения сопротивлений обычно выбирают довольно большими, чтобы пренебречь погрешностями, вносимыми конечным сопротивлением электронных ключей. Следует отметить, что схема (рис. 5.1, б) обладает более низким быстродействием, чем схема (рис. 5.1, а), так как содержит больше паразитных емкостей и в ней используется многозвенный принцип передачи напряжения. Этот недостаток схемы с резистивной цепью типа R-2R является причиной того, что схеме с двоично взвешенной резистивной цепью нередко отдается предпочтение в практических применениях.

Для обеспечения более высокой скорости преобразования на практике используются схемы ПКН, основанные на принципе суммирования токов (рис. 5. 2).

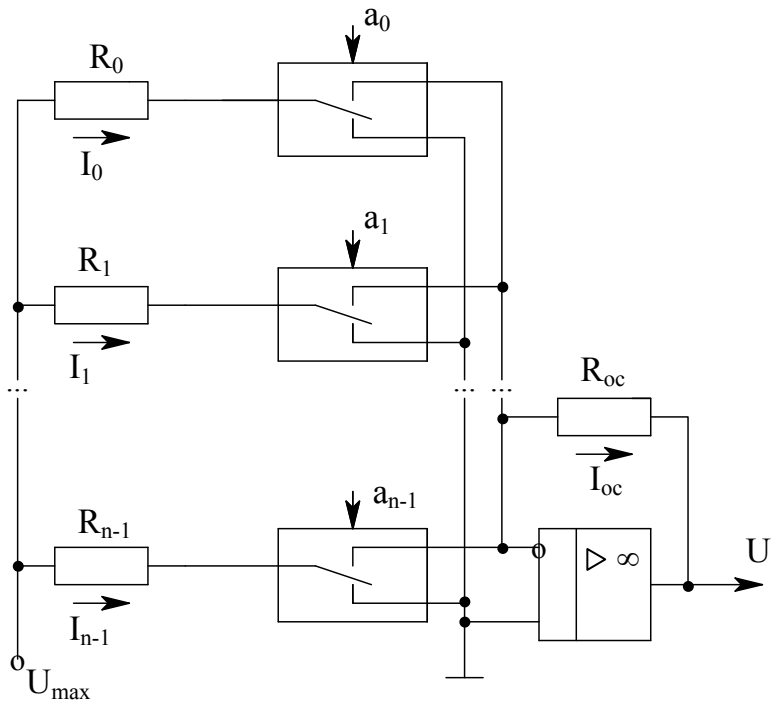


Рис. 5.2. Схема ПКН с суммированием токов

Такая схема отличается от рассмотренных расположением ключевых элементов и наличием ОУ с отрицательной обратной связью. Благодаря большим значениям коэффициента усиления и входного сопротивления ОУ, можно считать, что напряжение на его входе и входной ток ОУ близки к нулю. Следовательно все токи i_k, текущие через элементы резистивной цепи, уравновешиваются током I_{oc}, текущем в цепи обратной связи, и выходное напряжение ПКН равно

$$U(N) = R_{oc}\, I_{oc} = R_{oc} \sum_{k=0}^{n-1} a_k i_k \,, \qquad (5.4)$$

т. е. операционный усилитель выполняет операцию суммирования токов, которые определяются значениями сопротивлений в тех разрядах ПКН, где a = 1. Значения сопротивлений в схеме выбирают согласно правилу

$$R_k = R_{n-1}\, 2^{n-k-1}.$$

При этом

$$U(N) = R_{oc} \sum_{k=0}^{n-1} a_k \frac{U_{max}}{R_k} = \frac{2R_{oc}}{R_{n-1}} U_{max} \sum_{k=0}^{n-1} a_k \frac{1}{2^{n-k}}, \qquad (5.5)$$

т. е. выполняемая этой схемой операция есть операция преобразования двоичного кода в напряжение, описанное с помощью выражений (5.2 – 5.3).

Преимуществом схемы ПКН, использующей принцип суммирования токов, является ее более высокое быстродействие. Это объясняется тем, что в этой схеме токи, проходящие через резисторы, практически не меняются при изменении состояния ключей, так как входные напряжения и ток операционного усилителя близки к нулю, а это означает, что паразитные емкости при переключениях ключей не перезаряжаются.

5.2. Аналого-цифровые преобразователи

В основе построения преобразователей напряжение-код (ПНК) лежат в основном три известных принципа преобразования, определяющих алгоритм функционирования и структуру соответствующих устройств последовательного счета, поразрядного кодирования, считывания. Согласно этой классификации рассмотрим три основные разновидности схем ПНК.

Принцип последовательного счета реализуется с помощью схемы, представленной на рис. 5.3.

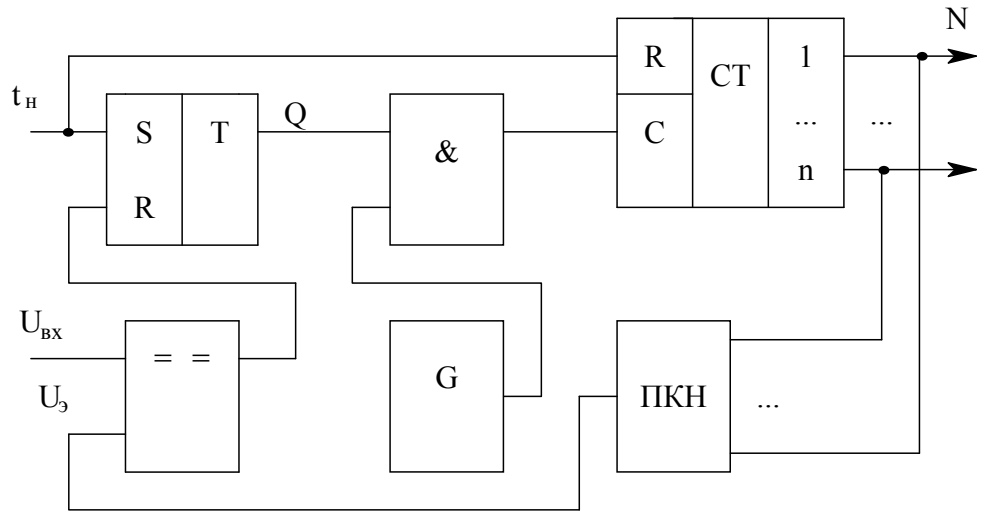


Рис. 5.3. Функциональная схема АЦП последовательного счета

Он состоит в сравнении входного напряжения $U_{вх}$ с последовательно нарастающим эталонным напряжением $U_э(t)$, представляющим собой сумму «квантов» ΔU, которые определяют погрешность преобразования.

Ступенчатое напряжение $U_э(t)$ формируется с помощью ПКН и двоичного счетчика СТ, последовательно изменяющего свое состояние, начиная от момента обнуления импульсом $t_н$, соответствующим началу операции преобразования. В момент совпадения эталонного напряжения $U_э(t_k) = N\ \Delta U$ с входным напряжением $U_{вх}$ (с точностью до «кванта» ΔU) схема сравнения вырабатывает импульс, поступающий на вход R триггера Т, и останавливающий счетчик путем подачи запрещающего сигнала с выхода Q триггера на схему совпадений, пропускающую на счетчик импульсы тактового генератора G. Этот момент времени соответствует окончанию операции преобразования (импульсы t_k) на временной диаграмме работы схемы (рис. 5.4).

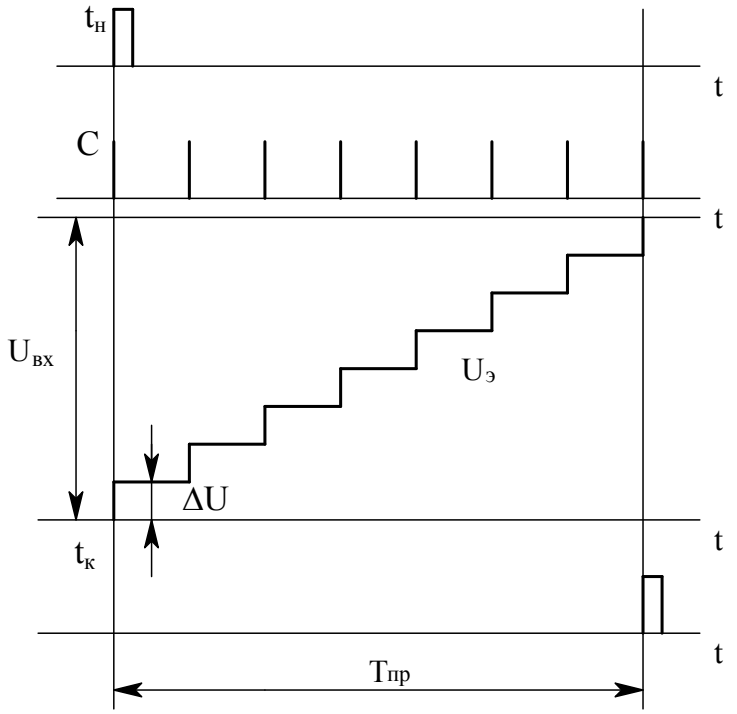


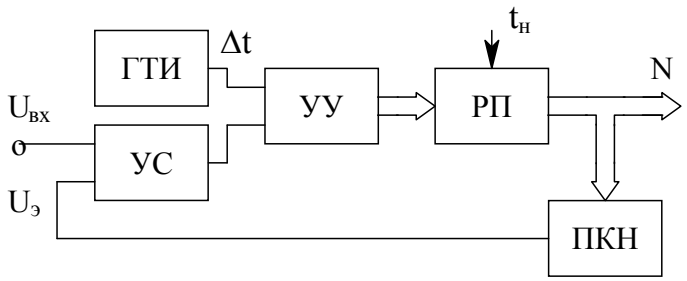
Рис. 5.4. Временная диаграмма работы АЦП последовательного счета

Погрешность преобразования в такой схеме определяется значением ΔU, которое, в свою очередь, зависит от точности ПКН и чувствительности схемы сравнения. Диапазон преобразования определяется разрядностью счетчика и ПКН, т. к. максимально допустимое значение входного напряжения $U_{max} = \Delta U(2^n - 1) \approx \Delta U 2^n$.

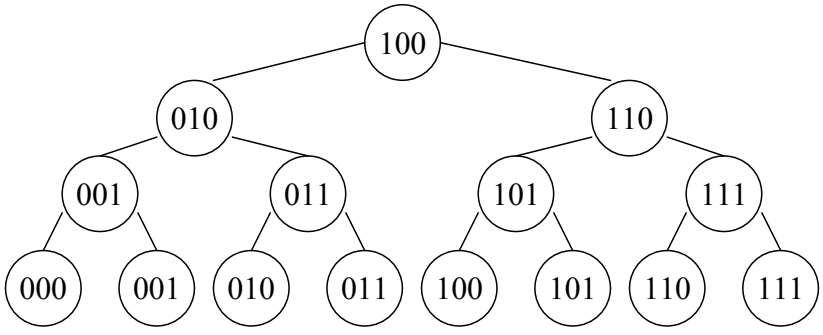
Основным недостатком ПНК, использующих принцип последовательного счета, является сравнительно большое время преобразования, зависящее от входного напряжения, быстродействия счетчика и ПКН. В предельном случае, когда входное напряжение максимально, т. е. $U_{вх} = \Delta U 2^n$, а время переключения счетчика Δt, полное время преобразования $T_{пр}$, характеризующее быстродействие ПНК, определяется как $T_{пр} = \Delta t 2^n$.

Указанный недостаток в значительной мере компенсируется тем, что подобные схемы являются наиболее простыми из всех возможных разновидностей ПНК. Поэтому такие ПКН широко применяются на практике, когда требование быстродействия не является определяющим.

Принцип *поразрядного кодирования* состоит в формировании цифровым способом эталонного напряжения $U_э$ путем последовательного приближения его к входному напряжению $U_{вх}$. Этот принцип поясняется структурной схемой (рис. 5.5, а) и графом переходов (рис. 5.5, б), который отражает алгоритм управления состояния регистра памяти (РП), необходимый для реализации метода последовательного приближения эталонного напряжения, снимаемого с выхода ПКН.



а



б

Рис. 5.5. Структурная схема (а) и граф переходов (б) АЦП поразрядного кодирования

Узлы графа характеризуют состояние регистра РП, т. е. содержащееся в нем двоичное число в каждый момент сравнения напряжений $U_э$ и $U_{вх}$. Направление перехода задается устройством управления (УУ) в зависимости от выходного сигнала устройства сравнения (УС). В начальный момент времени $t_н$ (момент запуска схемы) регистр устанавливается в состояние 10 ... 0 (100 – для 3-разрядного АЦП), при котором значение $U_э$ определяется весом старшего разряда выходного двоичного числа N. Затем осуществляется n тактов последовательного приближения напряжении $U_э$ к значению входного напряжения $U_{вх}$.

На каждом такте возможно два исхода, два управляющих воздействия, изменяющих состояния регистра (см. обозначения на ветвях графа): если $U_э < U_{ВХ}$, то производится установка очередного младшего разряда в «1» при сохранении состояния всех предшествующих старших разрядов; если $U_э > U_{ВХ}$, то установка младшего разряда в «1» сопровождается сбросом предыдущего старшего разряда в «0». В результате, по истечении n тактов управления (где n-число разрядов регистра), эталонное напряжение $U_э$ будет приближено к $U_{вх}$ с точностью до вклада самого младшего разряда, т. е.

$$U_Э = \sum_{i=0}^{n-1} a_i U_{эi} \approx U_{ВХ},$$

где $U_{эi} = U_{max}/2^{n-i}$ – вклад i-го разряда в напряжение $U_э$ на выходе ПКН, причем U_{max} характеризует максимальное преобразуемое напряжение; $U_{э0} = U_{max}/2^n$ – погрешность преобразования (т. е. вклад младшего разряда).

Преимуществом рассмотренного принципа преобразования по сравнению с принципом последовательного счета является значительно меньшее время преобразования, которое в данном случае определяется как $T_{пр} = n\,\Delta t$, где Δt – длительность одного такта управления, задаваемая генератором тактовых импульсов ПНК, которое заключено в устройстве управления. Функциональная схема ПНК с поразрядным кодированием представлена на рис. 5.6.

В этой схеме последовательная установка разрядов основного регистра памяти в единичное состояние осуществляется с помощью управляющего регистра сдвига, на вход которого записывается «1» в момент начала преобразования $t_н$.

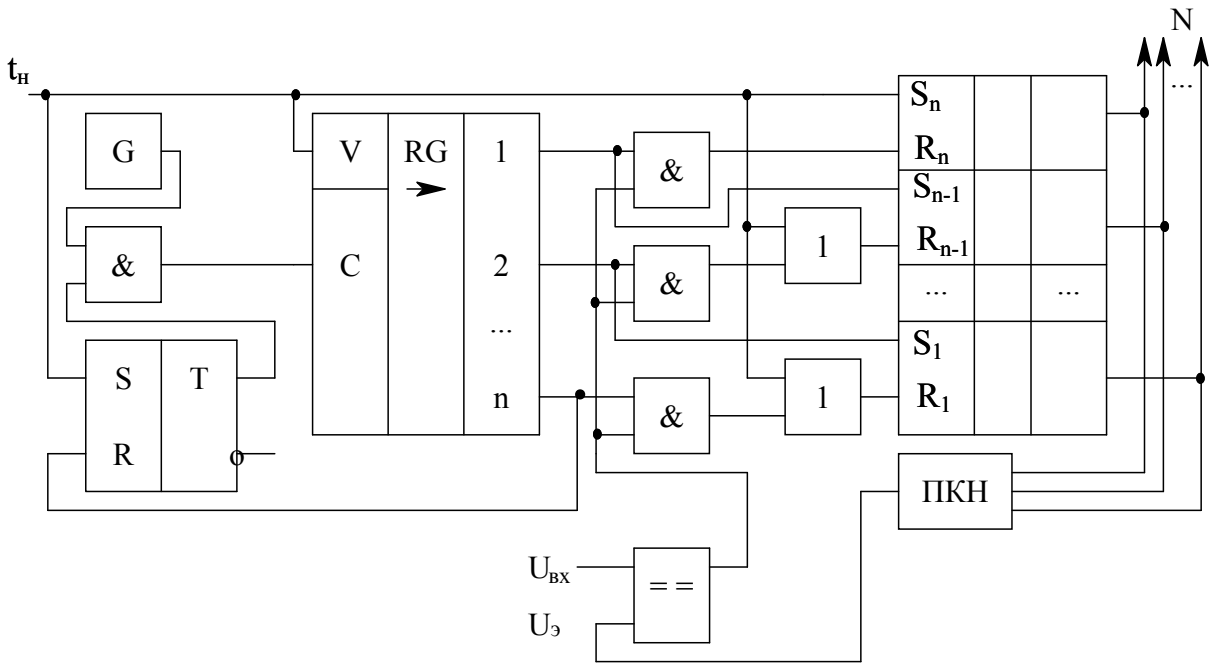


Рис. 5.6. Функциональная схема АЦП поразрядного кодирования

В этот же момент происходит установка в «1» старшего разряда основного регистра и обнуление всех остальных его разрядов. На каждом такте преобразования (задаваемом генератором G) сигнал с выхода схемы сравнения, вырабатываемый в случае превышения эталонным напряжением входного (т. е. при $U_э > U_{вх}$), проходит только через ту схему совпадений (И), на второй вход которой подается разрешающий сигнал с одного из разрядов сдвигающего регистра. При этом происходит сброс соответствующего разряда основного регистра в нулевое состояние. Затем происходит сдвиг разрешающего сигнала на вход следующего (младшего) разряда, который в момент переключения этого сигнала (т. е. в момент сдвига) устанавливается в 1. Если на данном такте управления $U_{вх} > U_э$, то сигнал сброса не вырабатывается схемой сравнения и соответствующий разряд регистра остается в единичном состоянии. В течение n тактов происходит последовательная установка в «1» или «0» всех разрядов регистра в соответствии с графом переходов. Операция преобразования заканчивается в момент перехода управляющего единичного сигнала в последний разряд сдвигающего регистра, так как при этом триггером Т вырабатывается сигнал запрета на схему совпадений, пропускающую тактовые импульсы на вход регистра сдвига.

Двоичное число N, пропорциональное входному напряжению $U_{вх}$, снимается в виде параллельного кода (a_{n-1} a_{n-2} … a_1 a_0) с выходного регистра после окончания преобразования.

Повысить скорость преобразования в ПНК можно также, используя параллельный набор возможных значений эталонного напряжения вместо их последовательного чередования, характерного для обоих рассмотренных принципов преобразования. Это приводит к так называемому параллельному принципу преобразования, именуемому также *принципом считывания*.

По принципу считывания схема ПНК (рис. 5.7) содержит m резистивных делителей эталонного напряжения и столько же схем сравнения. Число m определяется количеством дискретных значений преобразуемого напряжения в полном диапазоне преобразования, т. е. если максимальное значение напряжения $U_э$, а допустимая погрешность преобразования ΔU, то $m = U_э/\Delta U - 1$. Напряжения эталонных делителей удовлетворяют соотношению $U_{эi} = U_i/(m+1) = \Delta U_i$ для всех $i = [1, m]$. Следовательно, если напряжение $U_{вх}$ превышает значение $U_{эi}$, то происходит срабатывание тех схем сравнения, на которые подаются эталонные напряжения $U_{эi}$, $U_{эi-1}$, ... , $U_{э1}$, так как всегда $U_{эi} > U_{эi-1} > ... > U_{э1}$. Выходные сигналы схем сравнения устанавливают в единичное состояние соответствующие элементы m-разрядного запоминающего регистра RG (предварительно все элементы регистра устанавливаются в 0 в момент начала преобразования $t_н$).

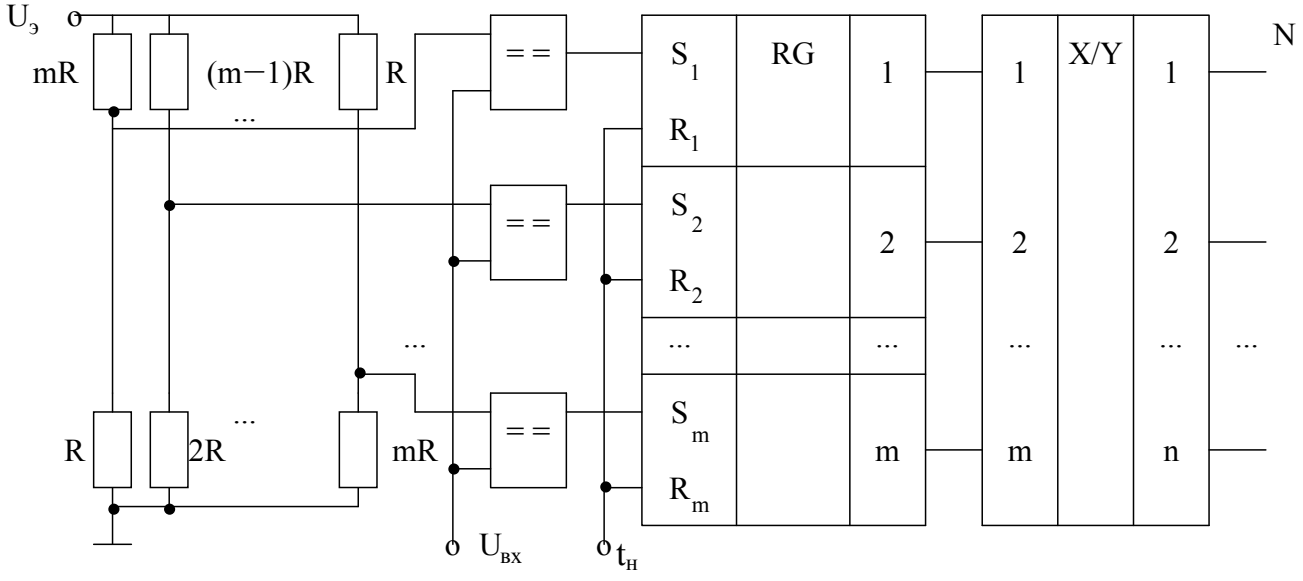


Рис. 5.7. Функциональная схема АЦП по методу считывания

Таким образом, в регистре RG образуется число, пропорциональное $U_{вх}$, в виде единичного позиционного кода. Обычно выполняется операция преобразования этого кода в параллельный двоичный код. Для этого в схему включен кодовый преобразователь, осуществляющий преобразование m-разрядного единичного кода в n-разрядный двоичный код при условии $m = 2^n - 1$.

Рассмотренный параллельный преобразователь обладает теоретически предельным быстродействием, так как входное напряжение за один шаг преобразования сравнивается с полным набором эталонных дискретных значений на всем интервале изменения преобразуемого напряжения. Длительность такой операции преобразования определяется временем срабатывания сравнивающих устройств и быстродействием цифровых элементов, составляющих регистр и кодовый преобразователь X/Y.

Однако это преимущество параллельных ПНК достигается ценой больших аппаратурных затрат, так как количество прецизионных делителей напряжения, схем сравнения и элементов памяти в таких ПНК растет пропорционально 2^n, если n – число двоичных разрядов преобразователя. Поэтому на практике параллельный принцип преобразования применяется только для построения быстродействующих ПНК малой разрядности.

Контрольные вопросы

1. Определить соотношение быстродействия ЦАП с двоично-взвешенными резистивными цепями и ЦАП с резистивной цепью типа R-2R.

2. Почему ЦАП по методу суммирования токов имеет наибольшее быстродействие?

3. Во сколько раз можно повысить быстродействие АЦП последовательного счета и поразрядного кодирования при уменьшении числа разрядов с 12 до 8?

4. Почему АЦП по методу считывания имеет сравнительно небольшое число разрядов?

ЧАСТЬ 2. ПРИНЦИПЫ ПОСТРОЕНИЯ КОМПЬЮТЕРОВ

6. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ КОМПЬЮТЕРОВ

6.1. Этапы развития ЭВМ

Развитие электронных вычислительных машин (ЭВМ) можно условно разбить на несколько этапов (поколений), которые имеют свои характерные особенности.

Первый этап (ЭВМ первого поколения) — до конца 1950-х гг.

Точкой отсчета эры ЭВМ считают 1946 г., когда был создан электронный цифровой компьютер – ENIAK (Electronic Numerical Integrator and Computer), созданный Джоном Моучли и Преспером Эккертом. Первая ЭВМ содержала 18000 электронных ламп, ее энергопотребление составляло 150 кВт. Вычислительные машины этого поколения строились на электронных лампах, потребляющих огромное количество электроэнергии и выделяющих много тепла.

Числа в ЭВМ вводились с помощью перфокарт и набора переключателей, а программа задавалась соединением гнезд на специальных наборных платах. Производительность такой гигантской ЭВМ была ниже, чем современного калькулятора. Широкому использованию этих ЭВМ препятствовали также низкая надежность, ограниченность их ресурсов и чрезвычайно трудоемкий процесс подготовки, ввод и отладка программ, написанных на языке машинных команд. Основными их пользователями были ученые, решавшие наиболее актуальные научно-технические задачи, связанные с развитием реактивной авиации, ракетостроения и т. д.

Среди известных отечественных машин первого поколения необходимо отметить БЭСМ-1, «Стрела», «Урал», М-20. Характеристики ЭВМ первого поколения (на примере БЭСМ-1, 1953 г.): емкость памяти — 2048 слов; быстродействие — от 7000 до 8000 опер./с; разрядность — 39 разрядов; арифметика — двоичная с плавающей запятой; система команд — трехадресная; устройство ввода — перфолента; количество электронных ламп в аппаратуре — около 4000; внешние запоминающие устройства — барабаны на 5120 слов; магнитная лента — до 120 000 слов; вывод на

быструю цифровую печать — 300 строк в минуту. Отечественная ЭВМ М-20 (20 тыс. опер./с) была одной из самых быстродействующих машин первого поколения в мире.

Основной режим использования ЭВМ первого поколения состоял в том, что математик, составивший программу, садился за пульт управления ЭВМ и производил необходимые вычисления. Чаще всего работа за пультом была связана с отладкой своей собственной программы — наиболее длительным по времени процессом. При этом уровень математика-программиста определялся его умением быстро находить и исправлять ошибки в своих программах, хорошо ориентироваться за пультом ЭВМ. В этот период началась интенсивная разработка средств автоматизации программирования, создание входных языков разных уровней, систем обслуживания программ, упрощающих работу на ЭВМ и увеличивающих эффективность ее использования.

Второй этап (ЭВМ второго поколения) — до середины 1960-х гг.

Развитие электроники привело к изобретению в 1948 г. нового полупроводникового устройства — транзистора, который заменил лампы. Создатели транзистора — сотрудники американской фирмы Bell Laboratories, физики У. Шокли, У. Браттейн и Дж. Бардин за это достижение были удостоены Нобелевской премии. Появление ЭВМ, построенных на транзисторах, привело к уменьшению их габаритов, массы, энергопотребления и стоимости, а также к увеличению их надежности и производительности. Одной из первых транзисторных ЭВМ была созданная в 1955 г. бортовая ЭВМ для межконтинентальной баллистической ракеты ATLAS.

Если с технической точки зрения переход к машинам второго поколения четко очерчен переходом на полупроводники, то со структурной точки зрения ЭВМ второго поколения характеризуются расширенными возможностями по вводу-выводу, увеличением емкости запоминающих устройств, развитыми системами программирования.

В рамках второго поколения все более четко проявляется разделение ЭВМ на малые, средние и большие, позволившие существенно расширить сферу применения ЭВМ, приступить к созданию автоматизированных систем управления (АСУ) предприятиями, целыми отраслями и технологическими процессами.

Стиль использования ЭВМ второго поколения характерен тем, что теперь математик-программист не допускался в машинный зал, а свою программу, обычно записанную на языке высокого уровня, отдавал в группу обслуживания, которая занималась дальнейшей обработкой его задачи — перфорированием и пропуском на ЭВМ.

Большой вклад в развитие вычислительной техники внес советский конструктор Сергей Александрович Лебедев. С 1951 г. под его руководством была создана первая в СССР ЭВМ – малая электронно-счетная машина.

Среди известных отечественных ЭВМ второго поколения необходимо отметить БЭСМ-4, М-220 (200 тыс. опер./с), «Наири», «Мир», «Минск», «Раздан», «Днепр». Наилучшей отечественной ЭВМ второго поколения считается БЭСМ-6, созданная в 1966 г. Она имела основную и промежуточную память (на магнитных барабанах) объемами, соответственно, 128 и 512 Кбайт, быстродействие порядка 1 млн опер./с и довольно обширную периферию (магнитные ленты и диски, графопостроители, разнообразные устройства ввода-вывода).

В этот период появились так называемые алгоритмические *языки высокого уровня,* средства которых допускают описание всей необходимой последовательности вычислительных действий в наглядном, легко воспринимаемом виде. Программа, написанная на алгоритмическом языке, непонятна компьютеру, воспринимающему только язык своих «собственных» машинных команд. Поэтому специальные программы, которые называются *трансляторами,* переводят программу с языка высокого уровня в машинный код.

Появился широкий набор библиотечных программ для решения разнообразных математических задач. Были созданы *мониторные системы,* управляющие режимом трансляции и исполнения программ. Из мониторных систем в дальнейшем получили развитие современные *операционные системы* (ОС) – комплексы служебных программ, обеспечивающих лучшее распределение ресурсов ЭВМ при исполнении пользовательских задач.

Первые ОС автоматизировали работу оператора ЭВМ, связанную с выполнением задания пользователя: ввод в ЭВМ текста программы, вызов нужного транслятора, вызов необходимых библиотечных программ и т. д. Теперь же вместе с программой и данными в ЭВМ вводится инструкция, где перечисляются этапы обработки и приводится ряд сведений о программе и ее авторе. Затем в ЭВМ стали вводить сразу по нескольку заданий пользователей (пакет заданий), ОС стали распределять ресурсы ЭВМ между этими заданиями — появился мультипрограммный режим обработки.

Третий этап (ЭВМ третьего поколения) — до начала 1970-х гг.

Элементной базой в ЭВМ третьего поколения являются интегральные схемы. Создание технологии производства интегральных схем, состоящих из десятков электронных элементов, образованных в прямоугольной пластине кремния с длиной стороны не более 1 см, позволило увеличить быстродействие и надежность ЭВМ на их основе, а также уменьшить габариты, потребляемую мощность и стоимость ЭВМ.

Машины третьего поколения — это семейство машин с единой архитектурой, т. е. программно-совместимых. Они имеют развитые операционные системы, обладают возможностями мультипрограммирования, т. е. одновременного выполнения нескольких программ.

Примеры ЭВМ третьего поколения — семейство IBM-360, IBM-370, PDP-8, PDP-11, отечественные ЕС ЭВМ (единая система ЭВМ), СМ ЭВМ (семейство малых ЭВМ) и др.

Быстродействие компьютеров изменяется от нескольких десятков тысяч до миллионов операций в секунду, емкость оперативной памяти достигает нескольких сотен тысяч слов.

В этот период широкое распространение получило семейство мини-ЭВМ. Простота обслуживания мини-ЭВМ, их сравнительно низкая стоимость и малые габариты позволили использовать их коллективами исследователей, разработчиками экспериментаторами и т. д., т. е. непосредственно пользователями. В начале 1970-х гг. с термином мини-ЭВМ связывали уже два существенно различных типа средств вычислительной техники:

– универсальный блок обработки данных и выдачи управляющих сигналов, серийно выпускаемых для применения в различных специализированных системах контроля и управления;

– универсальную ЭВМ небольших габаритов, проблемно-ориентированную пользователем на решение ограниченного круга задач, – микроЭВМ.

В период ЭВМ третьего поколения произошел крупный сдвиг в области применения компьютеров. Если раньше ЭВМ использовались в основном для научно-технических расчетов, то в 1960 – 1970-е гг. все больше места стала занимать обработка символьной информации.

Четвертый этап (ЭВМ четвертого поколения) — по настоящее время.

Этот этап условно делят на два периода: первый — до конца 1970-х гг. и второй — с начала 1980-х гг. по настоящее время.

В первый период успехи в развитии электроники привели к созданию больших интегральных схем (БИС), где в одном кристалле размещалось несколько десятков тысяч электронных элементов. Это позволило разработать ЭВМ, имеющие большие объемы памяти и скорости выполнения команд. Разрабатывались новые ОС, позволяющие программистам отлаживать свои программы прямо за дисплеем ЭВМ, что ускоряло разработку программ.

В 1971 г. был изготовлен первый микропроцессор — 4-разрядный Intel 4004 – большая интегральная схема (БИС), в которой полностью размещался процессор ЭВМ простой архитектуры. Стала реальной возможность размещения в одной БИС почти всех электронных устройств несложной по архитектуре ЭВМ. Появились микрокалькуляторы и микроконтроллеры — управляющие устройства, построенные на одной или нескольких БИС, содержащих процессор, память и системы связи с датчиками и исполнительными органами в объекте управления. Программа управления объектами вводилась в память ЭВМ либо при изготовлении, либо непосредственно на предприятии.

С появлением микропроцессоров начал развиваться новый класс компьютеров — *микроЭВМ.* МикроЭВМ подразделяются на несколько подклассов (рис. 6.1).

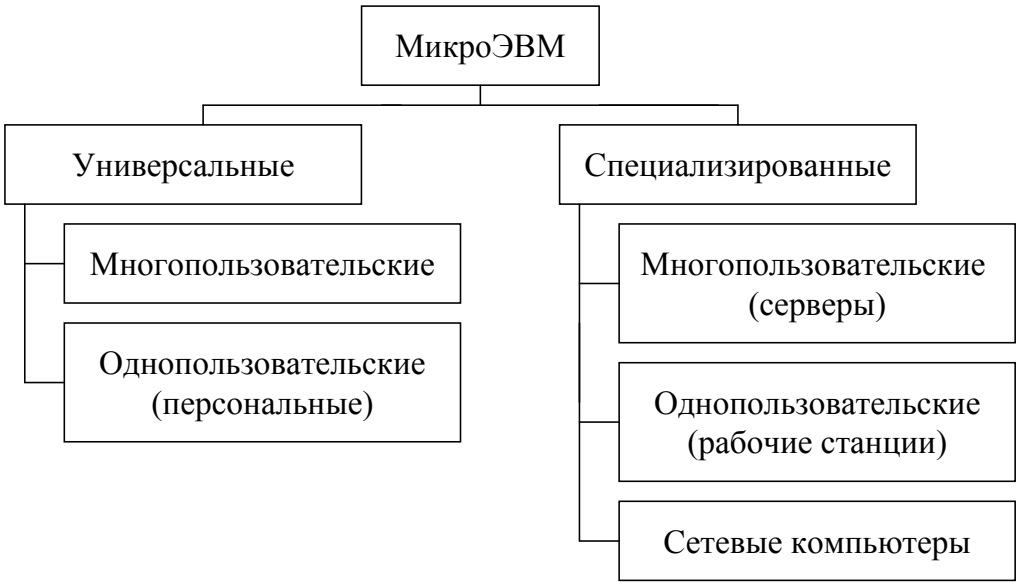


Рис. 6.1. Классификация микроЭВМ

Многопользовательские ЭВМ — это мощные микрокомпьютеры, оборудованные несколькими видеотерминалами и функционирующие в режиме разделения времени, что позволяет эффективно работать на них сразу нескольким пользователям.

Персональные компьютеры (ПК) — однопользовательские микрокомпьютеры, удовлетворяющие требованиям общедоступности и универсальности применения.

Рабочие станции (workstation) представляют собой однопользовательские микрокомпьютеры, часто специализированные для выполнения определенного вида работ (графических, инженерных, издательских и т. д.).

Серверы (server) — многопользовательские мощные микрокомпьютеры в вычислительных сетях, выделенные для обработки запросов от всех рабочих станций сети.

Сетевые компьютеры (network computer) — упрощенные микрокомпьютеры, обеспечивающие работу в сети и доступ к сетевым ресурсам, часто специализированные на выполнение определенного вида работ (защита сети от несанкционированного доступа, организация просмотра сетевых ресурсов, электронной почты и т. д.).

С точки зрения структуры ЭВМ этого поколения представляют собой многопроцессорные и многомашинные комплексы, работающие на общую память и общее поле внешних устройств. Для этого периода характерно широкое применение систем управления базами данных, компьютерных сетей, систем распределенной обработки данных.

Второй период четвертого поколения характеризуется улучшением технологии производства БИС, что позволило изготовлять электронные схемы, содержащие миллионы элементов в кристалле, схемы сверхбольшой степени интеграции (СБИС).

Перспективные поколения ЭВМ будут характеризоваться оптоэлектроникой с массовым параллелизмом и нейронной структурой — с распределенной сетью большого числа процессоров, моделирующих структуру нейронных биологических систем, произойдет качественный переход от обработки данных к обработке знаний.

6.2. Принципы фон Неймана

Со времени появления в 40-х гг. XX в. первых электронных цифровых вычислительных машин технология их производства была значительно усовершенствована. В последние годы благодаря развитию интегральной технологии существенно улучшились их характеристики. Однако несмотря на успехи, достигнутые в области технологии, существенных изменений в базовой структуре и принципах работы вычислительных машин не произошло. Так, в основу построения подавляющего большинства современных компьютеров положены общие принципы функционирования универсальных вычислительных устройств, сформулированные американским ученым *Джоном фон Нейманом* в 1945 году.

Согласно фон Нейману, для того чтобы ЭВМ была универсальным и эффективным устройством обработки информации, она должна строиться в соответствии со следующими принципами:

1. Информация кодируется в двоичной форме и разделяется на единицы (элементы) информации, называемые *словами*.

Использование в ЭВМ двоичных кодов продиктовано в первую очередь спецификой электронных схем, применяемых для передачи, хранения и преобразования информации. Как уже отмечалось, в этом случае конструкция ЭВМ предельно упрощается и ЭВМ работает наиболее надежно (устойчиво). Совокупности нолей и единиц (битов информации), используемые для представления отдельных чисел, команд и т. п., рассматриваются как самостоятельные информационные объекты и называются словами. Слово обрабатывается в ЭВМ как одно целое — как машинный элемент информации.

2. Разнотипные слова информации хранятся в одной и той же памяти и различаются по способу использования, но не по способу кодирования.

Все слова, представляющие числа, команды и прочие объекты, выглядят в ЭВМ совершенно одинаково и сами по себе неразличимы. Только порядок использования слов в программе вносит различия в них. Благодаря такому «однообразию» слов оказывается возможным использовать одни и те же операции для обработки слов различной природы, например для обработки и чисел, и команд, т. е. команды программы становятся в такой же степени доступными для отработки, как и числа.

3. Слова информации размещаются в ячейках памяти ЭВМ и идентифицируются номерами ячеек, называемыми адресами слов.

Структурно основная память состоит из перенумерованных ячеек. Ячейка памяти выделяется для хранения значения величины, в частности константы или команды. Чтобы записать слово в память, необходимо указать адрес ячейки, отведенной для хранения соответствующей величины. Чтобы выбрать слово из памяти (прочитать его), следует опять же указать адрес ячейки памяти, т. е. адрес ячейки, в которой хранится величина или команда, становится *машинным идентификатором* (именем) величины и команды. Таким образом, единственным средством для обозначения величин и команд в ЭВМ являются адреса, присваиваемые величинам и командам в процессе составления программы вычислений. При этом выборка (чтение) слова из памяти не разрушает информацию, хранимую в ячейке. Это позволяет любое слово, записанное однажды, читать какое угодно число раз, т. е. из памяти выбираются не слова, а копии слов.

4. Алгоритм представляется в форме последовательности управляющих слов, называемых командами, которые определяют наименование операции и слова информации, участвующие в операции. Алгоритм, представленный в терминах машинных команд, называется *программой*.

5. Выполнение вычислений, предписанных алгоритмом, сводится к последовательному выполнению команд в порядке, однозначно определяемом программой.

Первой выполняется команда, заданная пусковым адресом программы. Обычно это адрес первой команды программы. Адрес следующей команды однозначно определяется в процессе выполнения текущей команды и может быть либо адресом следующей по порядку команды, либо адресом любой другой команды. Процесс вычислений продолжается до тех пор, пока не будет выполнена команда, предписывающая прекращение вычислений.

Перечисленные принципы функционирования ЭВМ предполагают, что компьютер должен иметь следующие устройства:

– *арифметико-логическое устройство* (АЛУ), выполняющее арифметические и логические операции;

– *устройство управления* (УУ), которое организует процесс выполнения программы;

– *запоминающее устройство* (ЗУ), или память для хранения программ и данных;

– *внешние устройства* для ввода (устройства ввода) и вывода (устройства вывода) информации.

При рассмотрении компьютерных устройств принято различать их архитектуру и структуру.

Под ***архитектурой ЭВМ*** понимают ее логическую организацию, состав и назначение ее функциональных средств, принципы кодирования и т. п., т. е. все то, что однозначно определяет процесс обработки информации на данной ЭВМ. ЭВМ, построенные в соответствии с принципами фон Неймана, называют фоннеймановскими, или компьютерами фоннеймановской (классической) архитектуры.

Структура ЭВМ – совокупность элементов компьютера и связей между ними.

Ввиду большой сложности современных ЭВМ принято представлять их структуру иерархически, т. е. понятие «элемент» жестко не фиксируется. Так, на самом высоком уровне сама ЭВМ может считаться элементом. На следующем (программном) уровне иерархии элементами структуры ЭВМ являются память, процессор, устройства ввода-вывода и т. д. На более низком уровне (микропрограммном) элементами служат узлы и блоки, из которых строятся память, процессор и т. д. Наконец, на самых низких уровнях элементами выступают интегральные логические микросхемы и электронные приборы.

6.3. Принципы построения компьютеров

Основным принципом построения всех современных компьютеров является *программное управление.* В основе его лежит представление алгоритма решения любой задачи в виде программы вычислений. Все вычисления, предписанные алгоритмом решения задачи, должны быть представлены в виде программы, состоящей из последовательности управляющих слов — команд. Каждая команда содержит указания на конкретную выполняемую операцию, местонахождение (адреса) операндов и ряд служебных признаков. *Операнды* — переменные, значения которых участвуют в операциях преобразования данных.

Для доступа к программам, командам и операндам используются их адреса. В качестве адресов выступают номера ячеек памяти, предназначенных для хранения объектов. Информация кодируется двоичными цифрами «0» и «1». Поэтому различные типы информации, размещенные в памяти, практически не различимы, идентификация их возможна лишь при выполнении программы.

Обобщенная структурная схема компьютеров первых поколений представлена на рис. 6.2. В составе схемы имеются устройства ввода информации (УВв), с помощью которых пользователи вводят в компьютер программы решаемых задач и данные к ним. Сначала введенная информация запоминается в оперативном запоминающем устройстве (ОЗУ), а затем переносится во внешнее запоминающее устройство (ВЗУ), предназначенное для длительного хранения информации.

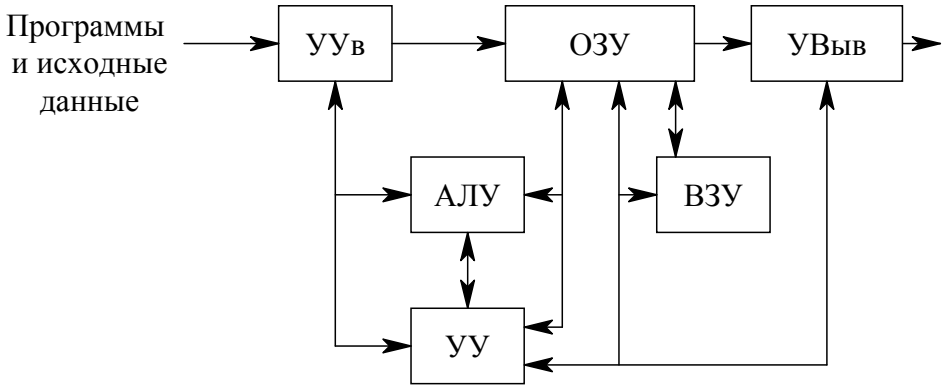


Рис. 6.2. Структурная схема компьютеров первого и второго поколений

Устройство управления (УУ) предназначается для автоматического выполнения программ путем принудительной координации работы всех устройств ЭВМ. Цепи сигналов управления показаны на рис. 6.2 штриховыми линиями. Вызываемые из ОЗУ команды дешифрируются устройством управления: определяется код операции, который необходимо выполнить, и адреса операндов, принимающих участие в данной операции.

Арифметико-логическое устройство (АЛУ) выполняет арифметические и логические операции над данными. Основной частью АЛУ является операционный блок, в состав которого входят сумматоры, счетчики, регистры, логические преобразователи и др. Результаты выполнения отдельных операций сохраняются для последующего использования в одном из регистров АЛУ или записываются в память. Результаты,

полученные после выполнения всей программы вычислений, передаются на устройства вывода (УВыв) информации. В качестве УВыв могут использоваться экран дисплея, принтер, графопостроитель и др.

В компьютерах третьего поколения произошло усложнение структуры за счет разделения процессов ввода-вывода информации и процесса ее обработки (рис. 6.3).

В структуре имеются устройства: управления обмена информацией (УВВ), каналы ввода-вывода (КВВ). КВВ получили наибольшее распространение применительно к большим ЭВМ. Здесь наметилась тенденция к децентрализации управления и параллельной работе отдельных устройств, что позволило повысить быстродействие компьютеров.

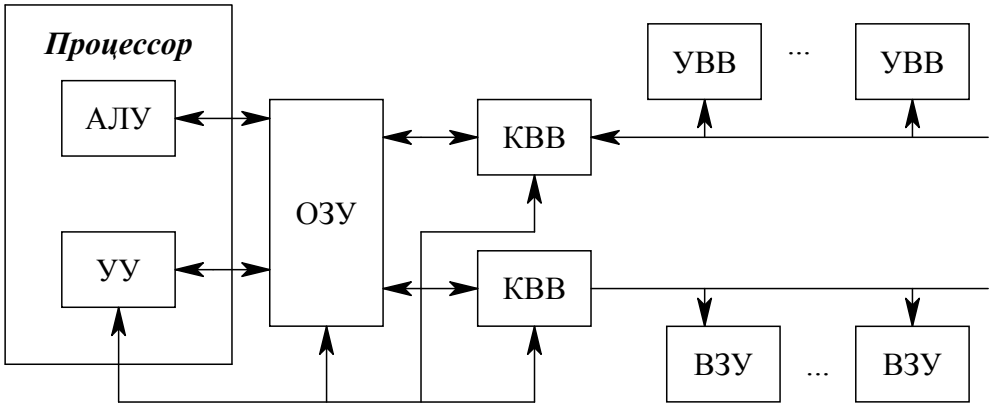


Рис. 6.3. Структурная схема компьютера третьего поколения

Среди каналов ввода-вывода выделяются мультиплексные каналы, способные обслуживать большое количество медленно работающих устройств ввода-вывода, и селекторные каналы, обслуживающие в монопольных режимах скоростные внешние запоминающие устройства (ВЗУ).

В ПЭВМ, относящихся к компьютерам *четвертого* поколения, произошло дальнейшее изменение структуры (рис. 6.4).

Соединение всех устройств обеспечивается с помощью общей шины, представляющей собой линии передачи данных, адресов, сигналов управления и питания. Единая система аппаратурных соединений значительно упростила структуру, сделав ее еще более децентрализованной. Все передачи данных по шине осуществляются под управлением сервисных программ.

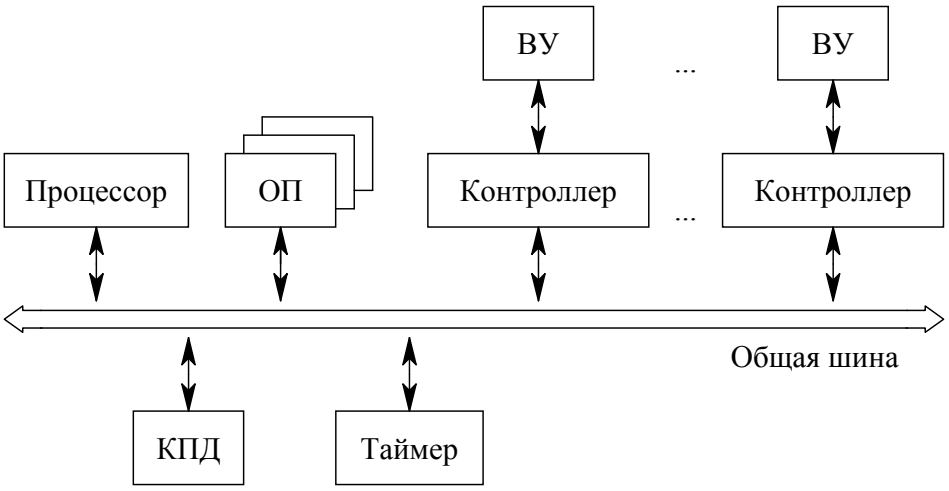


Рис. 6.4. Структурная схема ПЭВМ

Ядро ПЭВМ образуют процессор и основная память, состоящая из ОЗУ и постоянного запоминающего устройства (ПЗУ). ПЗУ предназначено для постоянного хранения программ управления. Подключение всех внешних устройств (ВУ): дисплея, клавиатуры, внешних ЗУ и др., обеспечивается через соответствующие адаптеры — согласователи скоростей работы сопрягаемых устройств или контроллеры — специальные устройства управления периферийной аппаратурой. Контроллеры в ПЭВМ играют роль каналов ввода-вывода. В качестве особых устройств следует выделить таймер — устройство измерения времени и контроллер прямого доступа к памяти (КПД) — устройство, обеспечивающее доступ к основной памяти, минуя процессор.

Все приведенные структуры не выходят за пределы классической структуры фон Неймана и их объединяют следующие общие принципы построения:

- структура памяти;
- способы доступа к памяти и внешним устройствам;
- система и форматы команд;
- организация интерфейса;
- возможность изменения конфигурации компьютера.

6.4. Центральный процессор

Процессор – устройство, непосредственно осуществляющее обработку информации и выполняющее функции управления работой ЭВМ в соответствии с заданной программой.

Процессор состоит из арифметико-логического устройства (АЛУ), устройства управления (УУ) и блока внутренних регистров (БВР) (рис. 6.5). Узлы соединены шинами, по которым передаются адреса – шина адреса (ША), данные – шина данных (ШД) и сигналы управления – шина управления (ШУ).

Арифметико-логическое устройство (АЛУ) предназначено для выполнения различных арифметических и логических операций над словами (операндами). *Арифметические операции АЛУ:* сложение, вычитание, умножение, деление кодов двоичных чисел с фиксированной и плавающей запятой в двоичной (шестнадцатеричной) и двоично-десятичной СС.

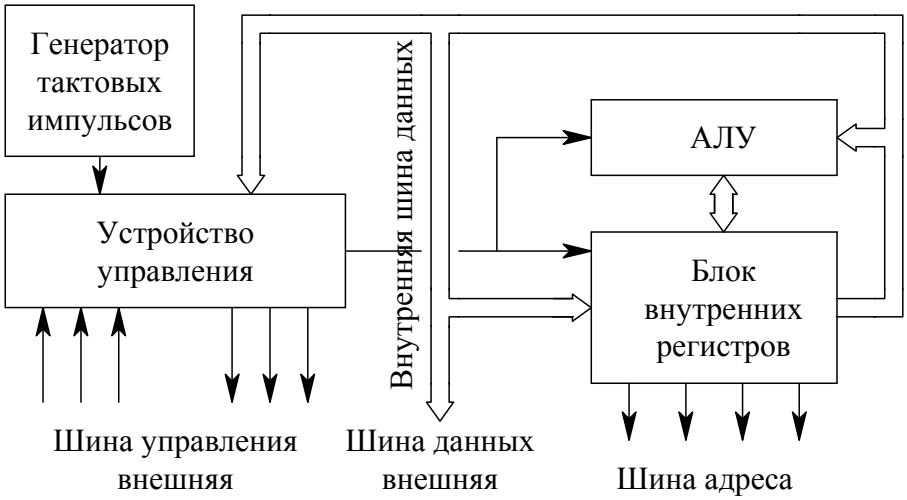


Рис. 6.5. Структура процессора

В АЛУ выполняются операции: нормализации числа, сдвига арифметического (без сдвига кода знака) и логического (сдвиг кода всего числа со знаком) и др. Кроме основных *логических операций* дизъюнкции (ИЛИ), конъюнкции (И) и инверсии (НЕ), в АЛУ предусматриваются другие операции, необходимые для логической обработки алфавитно-цифровой информации: сравнение слов на равенство, сортировка их по признакам и т. п. В отличие от арифметических операций, логические операции

являются *поразрядными:* они выполняются одновременно над одноименными разрядами слов независимо от значений соседних разрядов.

Классификация АЛУ

Все существующие АЛУ можно разбить на следующие классы:

1. По способу представления чисел различают АЛУ, в которых двоичные, шестнадцатеричные или (и) десятичные числа представляются в формах с фиксированной, плавающей или (и) естественной запятой.

2. По последовательности обработки разрядов операндов АЛУ делятся на последовательные и параллельные. В последовательных АЛУ операнды поступают последовательным кодом и их обработка (например, суммирование) производится разряд за разрядом последовательно во времени. В параллельных АЛУ операнды поступают параллельным кодом и их разряды обрабатываются одновременно.

3. По принципам построения основных (базовых) блоков различают АЛУ с комбинационным и накапливающим сумматором, АЛУ на однородных вычислительных средах (матричные), АЛУ с табличным выполнением операций и др.

4. По характеру использования блоков различают АЛУ со специализированными блоками (блочные) и АЛУ с многофункциональными блоками. В блочных АЛУ для выполнения каждой операции или каждой группы однотипных операций предусмотрены отдельные блоки. Например, блоки суммирования двоичных и десятичных чисел с фиксированной и плавающей точкой, блок умножения, блок деления, блок логических операций и другие. Такая специализация блоков сама по себе, а также возможность одновременной работы нескольких блоков, существенно увеличивает быстродействие АЛУ.

В многофункциональных АЛУ одни и те же узлы и блоки используются для выполнения различных операций. При настройке на конкретную операцию производится коммутация этих узлов в требуемую схему. Многофункциональные АЛУ более компактны, но их быстродействие ниже, чем у блочных АЛУ ввиду меньшей специализации и дополнительных задержек в узлах коммутации.

Основу АЛУ составляет операционный блок, на вход которого поступают данные (операнды). В операционном блоке над поступающими операндами производятся все операции, предусмотренные назначением АЛУ. Устройство управления вырабатывает последовательность сигналов управления под воздействием генератора

тактовых импульсов. Каждый сигнал управления поступает в операционный блок в определенный тактовый интервал и инициирует определенную микрооперацию над операндами (прием в регистр, сдвиг, инвертирование и т. п.). Каждая операция над операндами в АЛУ представляет собой определенную последовательность микроопераций.

Требуемая операция над операндами в АЛУ задается кодом операции. Этот код вызывает формирование в устройстве управления соответствующей последовательности сигналов управления.

Состав операционного блока определяется его предназначением — перечнем операций, формами представления операндов. Для выполнения операций над операндами в состав АЛУ входят (рис. 6.6): входные регистры А и В (РгА, РгВ) для приема операндов, n-разрядный сумматор (СМ), регистры для приема и хранения промежуточного и окончательного результата операции Рг S и Рг вых, блок логических операций – для формирования сигналов оповещения о промежуточных и окончательных значениях операндов (положительный или отрицательный результат, равен нулю или переполнение разрядной сетки и др.). Логические операции выполняются в регистрах с соответствующими логическими схемами или в блоке логических операций. Схема соединения узлов, указанных на рис. 6.6, меняется в зависимости от выполняемой операции.

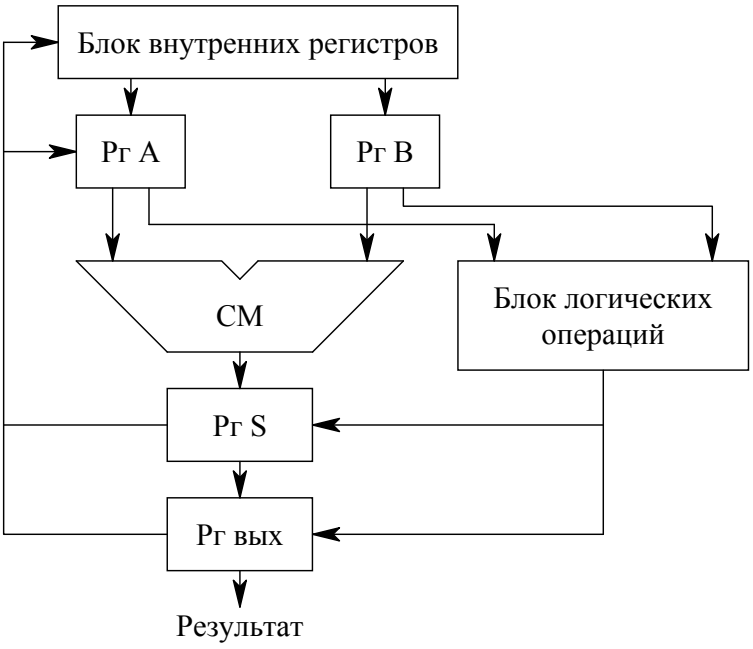


Рис. 6.6. Состав многофункционального АЛУ

АЛУ выполняет арифметические и логические операции над входными регистрами, результат операции помещается в один из входных регистров или блок внутренних регистров. Этот процесс называется циклом тракта данных.

УУ предназначено для организации автоматического выполнения заданной программы решения задачи. Это осуществляется путем принудительной координации работы всех устройств ЭВМ, т. е. посылкой в различные узлы и блоки управляющих сигналов в определенные моменты времени. Главная задача УУ – преобразование первичной командной информации, представленной программой решения задачи, во вторичную командную информацию, представленную управляющими сигналами. Воздействие этих сигналов на другие узлы и устройства ЭВМ обеспечивают автоматическое решение задачи.

УУ в составе ЭВМ выполняет следующие функции:

– управление последовательностью команд, т. е. определение адреса следующей команды в реализуемой программе, ее чтение и хранение на время выполнения;

– управление последовательностью операций и микроопераций в процессе выполнения команды;

– согласование, синхронизация работы узлов и устройств ЭВМ;

– контроль за работой ЭВМ и индикация состояний ее элементов.

Блок внутренних регистров служит внутренней памятью процессора и используется для временного хранения данных и команд, а также выполняет процедуры обработки информации.

6.5. Формы представления чисел в ЭВМ

Информация в памяти ЭВМ записывается в форме цифрового двоичного кода. С этой целью ЭВМ содержит большое количество ячеек памяти и регистров (от лат. *regestum* — внесенное, записанное) для хранения двоичной информации. Большинство этих ячеек имеет одинаковую длину *n,* т. е. они используются для хранения *n* бит двоичной информации (бит — один двоичный разряд). Информация, хранимая в такой ячейке, называется словом. Двоичное слово, состоящее из 2 байт, представлено на рис. 6.7.

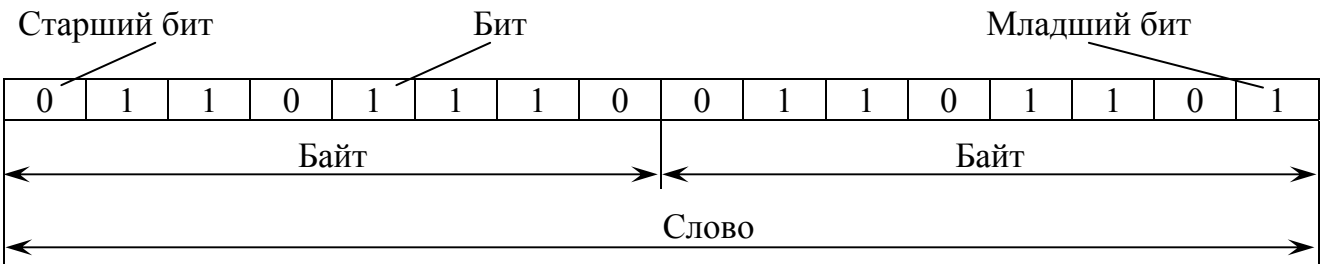


Рис. 6.7. Бит, байт и слово

Ячейки памяти и регистры состоят из элементов памяти. Каждый из таких электрических элементов может находиться в одном из двух устойчивых состояний: конденсатор заряжен или разряжен, транзистор находится в проводящем или непроводящем состоянии, специальный полупроводниковый материал имеет высокое или низкое удельное сопротивление и т. п. Одно из таких физических состояний создает высокий уровень выходного напряжения элемента памяти, а другое — низкий. Обычно это электрическое напряжение порядка 4—5 В и 0 В соответственно, причем первое обычно принимается за двоичную единицу, а второе — за двоичный ноль (возможно и обратное кодирование).

Память ЭВМ состоит из конечной последовательности слов, а слова — из конечной последовательности битов, поэтому объем представляемой в ЭВМ информации ограничен емкостью памяти, а числовая информация может быть представлена только с определенной точностью, зависящей от архитектуры памяти данной ЭВМ.

В ЭВМ применяются две формы представления двоичных чисел:

- естественная форма, или форма с фиксированной запятой (точкой);
- нормализованная форма, или форма с плавающей запятой (точкой).

С фиксированной запятой все числа изображаются в виде последовательности цифр с постоянным для всех чисел положением запятой (точкой), отделяющей целую часть от дробной.

Представление целых чисел и выполнение арифметических операций над ними

Целые числа могут представляться без знака или со знаком и занимают в памяти компьютера один или два байта (табл. 6.1).

В однобайтовом формате принимают значения от 00000000_2 до 11111111_2. В двухбайтовом формате — от 00000000 00000000_2 до 11111111 11111111_2.

Таблица 6.1

Формат числа в байтах	Диапазон	
	запись с порядком	обычная запись
1	$0 \ldots 2^8 - 1$	0... 255
2	$0 \ldots 2^{16} - 1$	0 ... 65535

Целые числа со знаком занимают в памяти компьютера один, два или четыре байта (табл. 6.2).

Таблица 6.2

Формат числа в байтах	Диапазон	
	запись с порядком	обычная запись
1	$-2^7 \ldots 2^7 - 1$	– 128 ... 127
2	$-2^{15} \ldots -2^{15} - 1$	– 32768 ... 32767
4	$-2^{31} \ldots 2^{31} - 1$	– 2147483648 ... 2147483647

В вычислительной технике применяются три формы записи (кодирования) целых чисел со знаком: прямой код, дополнительный код, обратный код.

Прямой код. Прямой n-разрядный двоичный код отличается от двоичного тем, что в нем отводится один, как правило, самый старший разряд для знака, а оставшиеся $n - 1$ разрядов — для значащих цифр. Значение знакового разряда равно 0 для чисел $A_2 > 0$, и 1 — для чисел $A_2 < 0$ (рис. 6.8).

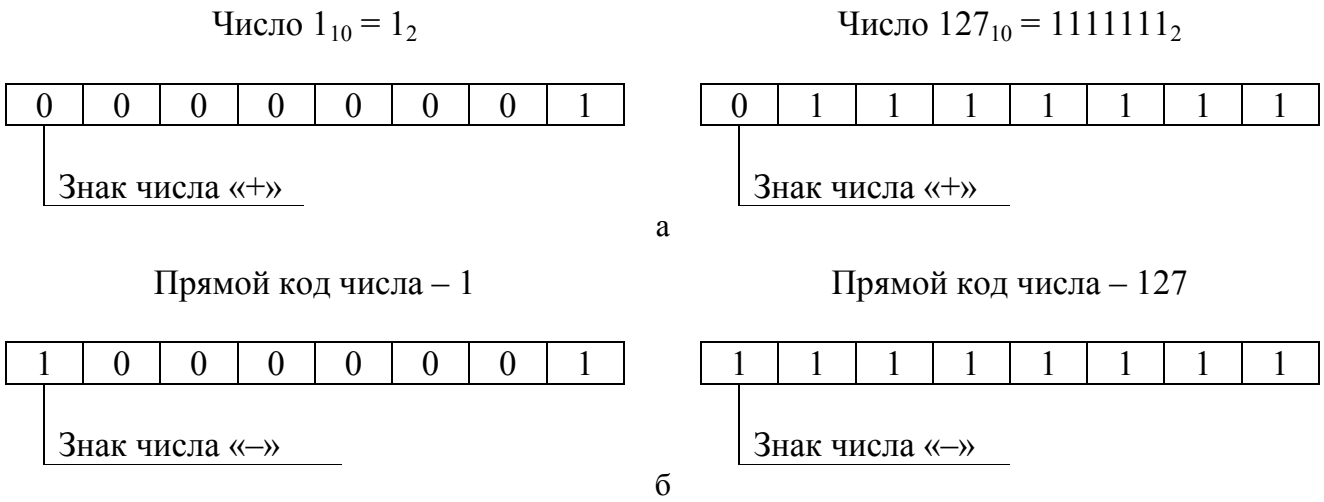


Рис. 6.8. Представление чисел в прямом коде:
а) положительных, б) отрицательных

Для прямого кода справедливо следующее соотношение:

$$A_{10} = (-1)^{a_{зн}} \sum_{i=0}^{n-2} a_i 2^i ,$$

где n – разрядность кода; $a_{зн}$ – значение знакового разряда.

Например, десятичная запись числа, представленного в прямом коде как 1101, будет иметь вид

$$A_{10} = (-1)^1 [1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2] = -5.$$

Дополнительный код. Использование чисел со знаком (прямого кода представления чисел) усложняет структуру ЭВМ. В этом случае операция сложения двух чисел, имеющих разные знаки, должна быть заменена на операцию вычитания меньшей величины из большей и присвоения результату знака большей величины. Поэтому в современных ЭВМ, как правило, отрицательные числа представляют в виде дополнительного или обратного кода, что при суммировании двух чисел с разными знаками позволяет заменить вычитание на обычное сложение и упростить тем самым конструкцию арифметико-логического устройства компьютера.

Обратный код. Для представления отрицательных чисел используется также обратный код, который получается инвертированием всех цифр двоичного кода абсолютной величины числа: нули заменяются единицами, а единицы — нулями. При этом необходимо помнить, что все операции с отрицательными числами выполняются в формате машинного слова. Это значит, что к двоичному числу слева дописываются нули до нужного количества разрядов. Например, для 8-разрядного машинного слова:

Число: – 1	Число: – 127
Код модуля числа: 00000001	Код модуля числа: 01111111
Обратный код числа: 11111110	Обратный код числа: 10000000

1	1	1	1	1	1	1	0

1	0	0	0	0	0	0	0

Для обратного кода справедливо следующее соотношение:

$$A_{10} = a_{зн}(-2^{n-1} + 1) + \sum_{i=0}^{n-2} a_i^0 2^i ,$$

где *n* — разрядность машинного слова.

$$a_{зн} = \begin{cases} 0, \text{ для положительных чисел} \\ 1, \text{ для отрицательных чисел} \end{cases}$$

Например, десятичная запись отрицательного числа, представленного в обратном коде как 1010, будет иметь вид

$$A_{10} = 1 \cdot (-2^3 + 1) + [0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2] = -7 + 2 = -5.$$

Для положительных чисел $a_{зн} = 0$ и представление числа полностью совпадает с представлением в прямом и дополнительном кодах.

Таким образом, положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково — двоичными кодами с цифрой 0 в знаковом разряде.

Обычно отрицательные десятичные числа при вводе в ЭВМ автоматически преобразуются в обратный или дополнительный двоичный код и в таком виде хранятся, перемещаются и участвуют в операциях. При выводе таких чисел из ЭВМ происходит обратное преобразование в отрицательные десятичные числа.

Выполнение арифметических операций над кодами

В большинстве вычислительных устройств с целью упрощения их конструкции операция вычитания не используется. Вместо нее производится сложение обратных или дополнительных кодов уменьшаемого и вычитаемого.

1. X и Y положительные. При суммировании складываются все разряды, включая разряд знака. Так как знаковые разряды положительных слагаемых равны нулю, разряд знака суммы тоже равен нулю. Например:

Десятичная запись	*Двоичные коды*
3	0 0000011
+ 7	+ 0 0000111
10	0 0001010

Получен корректный результат.

2. X положительное, Y отрицательное и по абсолютной величине больше, чем X.

Например:

Десятичная запись	*Двоичные коды*	
3	0 0000011	
+ –10	+ 1 1110101	Обратный код числа – 10
– 7	1 1111000	Обратный код числа – 7

Получен корректный результат в обратном коде. При переводе в прямой код биты цифровой части результата инвертируются: 1 0000111 = – 7_{10}.

3. *X положительное, Y отрицательное и по абсолютной величине меньше, чем X.*

Например:

Десятичная запись	*Двоичные коды*	
10	0 0001010	
+ – 3	+ 1 1111100	*Обратный код числа – 3*
7	0 0000110	
	→ +1	
	0 0000111	

Компьютер исправляет полученный первоначально некорректный результат (6 вместо 7) *переносом единицы* из знакового разряда в младший разряд суммы.

4. *X и Y отрицательные.*

Например:

Десятичная запись	*Двоичные коды*	
– 3	1 1111100	*Обратный код числа – 3*
+ – 7	+ 1 1111000	*Обратный код числа – 7*
– 10	1 1110100	
	→ +1	
	1 1110101	*Обратный код числа – 10*

Полученный первоначально некорректный результат (обратный код числа -11_{10} вместо обратного кода числа -10_{10}) компьютер исправляет переносом единицы из знакового разряда в младший разряд суммы. При переводе результата в прямой код биты цифровой части числа инвертируются: 1 0001010 = -10_{10}.

При сложении может возникнуть ситуация, когда старшие разряды результата операции не помещаются в отведенной для него области памяти. Такая ситуация называется переполнением разрядной сетки формата числа. Для обнаружения переполнения и оповещения о возникшей ошибке в компьютере используются специальные средства.

Представление вещественных чисел

С плавающей запятой каждое число изображается в виде двух групп цифр. Первая группа цифр называется мантиссой, вторая — порядком, причем абсолютная величина мантиссы должна быть меньше 1, а порядок — целым числом.

Любое вещественное число N, представленное в системе счисления с основанием *p,* можно записать в виде:

$N = \pm M \cdot p^{\pm k}$,

где M – мантисса; p – основание системы счисления;

k – порядок числа (целое число).

Например, десятичное число 234,47 или двоичное число 1011,01 можно представить следующим образом:

$234{,}47 = 234\,470{,}0 \cdot 10^{-3} = 23\,447{,}0 \cdot 10^{-2} = 2344{,}7 \cdot 10^{-1} = 234{,}47 \cdot 10^{0} =$

$= 23{,}447 \cdot 10^{1} = 2{,}3447 \cdot 10^{2} = 0{,}23447 \cdot 10^{3} = 0{,}023447 \cdot 10^{4}.$

$1011{,}01 = 1011{,}01 \cdot 10^{0} = 101{,}101 \cdot 10^{01} = 10{,}1101 \cdot 10^{10} = 1{,}01101 \cdot 10^{11} =$

$= 0{,}101101 \cdot 10^{100} = 0{,}0101101 \cdot 10^{101}.$

Такое представление чисел называется *представлением с плавающей запятой,* где порядок определяет, на сколько разрядов необходимо осуществить сдвиг относительно запятой.

Если плавающая запятая расположена в мантиссе перед первой значащей цифрой, то при фиксированном количестве разрядов, отведенных под мантиссу, обеспечивается запись максимального количества значащих цифр числа, т. е. максимальная точность представления числа. Из этого следует, что мантисса должна быть правильной дробью ($|M| < 1$), у которой первая цифра *a* после запятой отлична от нуля ($|M| = 0, a...$). Для двоичной системы счисления $a = 1$, поэтому $|M| = 0, 1...$. Если это требование выполнено, то число называется нормализованным. Нормализованная мантисса в двоичной системе счисления всегда представляется десятичным числом *n,* лежащим в диапазоне $0{,}5 \le n < 1$.

При написании вещественных чисел в программах вместо привычной запятой принято ставить точку.

Примеры нормализованного представления

Десятичная система *Двоичная система*

$753{,}15 = 0.75315 \cdot 10^{3}$; $-101{,}01 = -0.10101 \cdot 10^{11}$ (порядок $11_2 = 3_{10}$);

$-0{,}000034 = -0.34 \cdot 10^{-4}$; $0{,}000011 = 0.11 \cdot 10^{-100}$ (порядок $-100_2 = -4_{10}$).

Вещественные числа в компьютерах представляются в трех форматах: одинарном, двойном и расширенном, имеющих одинаковую структуру вида (рис. 6.9).

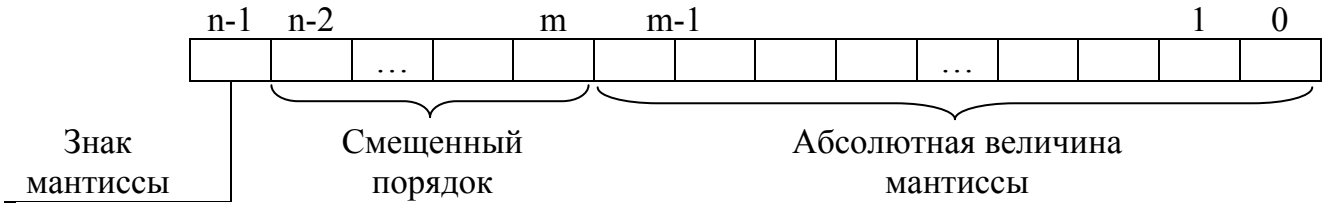


Рис. 6.9. Формат представления вещественных чисел

Здесь порядок n-разрядного нормализованного числа задается смещенным кодом, позволяющим производить операции над порядками как над беззнаковыми числами, что упрощает операции сравнения, сложения и вычитания порядков, а также операцию сравнения самих нормализованных чисел.

Чем больше разрядов отводится под запись мантиссы, тем выше точность представления числа. Чем больше разрядов занимает порядок, тем шире диапазон от наименьшего отличного от ноля числа до наибольшего числа, представимого в ЭВМ при заданном формате.

Контрольные вопросы

1. Назовите этапы развития ЭВМ и их характерные особенности.
2. В чем заключаются принципы фон Неймана?
3. Какие основные устройства входят в состав ЭВМ?
4. Дайте определение архитектуры и структуры ЭВМ.
5. Что лежит в основе программного управления?
6. Какие формы представления двоичных чисел применяются в ЭВМ?
7. Порядок представления отрицательных целых чисел в прямом, обратном и дополнительном кодах.
8. Сложить числа +10 и – 5, используя коды.

7. МИКРОЭВМ

7.1. Структурная схема микроЭВМ

МикроЭВМ – это проблемно-ориентированная ЭВМ, построенная на микропроцессорном комплекте БИС и предназначенная для решения задач автоматизации управления различными объектами. Типовая структурная схема микроЭВМ (рис. 7.1) состоит из микропроцессора, памяти и блока ввода-вывода.

В состав микропроцессора входят АЛУ, УУ, регистры и внутренние шины данных, управления и адреса.

АЛУ выполняет одну из главных функций – обработку данных, имеет два входных порта и один выходной. Каждый входной порт снабжен буферным регистром, способным хранить для АЛУ одно слово данных. Два входных порта позволяют АЛУ принимать данные или с внутренней шины данных микропроцессора или аккумулятора. Результат операции в АЛУ всегда помещается в аккумулятор. Перечень функций АЛУ зависит от типа МП. Типовыми операциями, выполняемыми АЛУ, являются: сложение, вычитание, И, ИЛИ, исключающее ИЛИ, инверсия, сдвиг вправо, сдвиг влево, приращение положительное, приращение отрицательное.

УУ предназначено для декодирования команды, находящейся в регистре команд, посредством дешифратора команд, который в результате выдает сигналы, необходимые для выполнения команды. УУ генерирует сигналы в соответствии с той или иной фазой синхросигналов и формируют в эти моменты времени соответствующие выходные сигналы для внешних устройств.

Регистры предназначены для реализации арифметических и логических функций МП и могут использоваться для временного хранения данных. Количество и назначение регистров в микропроцессоре зависят от его архитектуры. Одни регистры имеют специальное назначение, другие – многоцелевые. Регистры последнего типа называются регистрами общего назначения.

Регистры специального назначения в МП: аккумулятор, счетчик команд, регистр адреса, регистр команд, регистр состояния, буферные регистры.

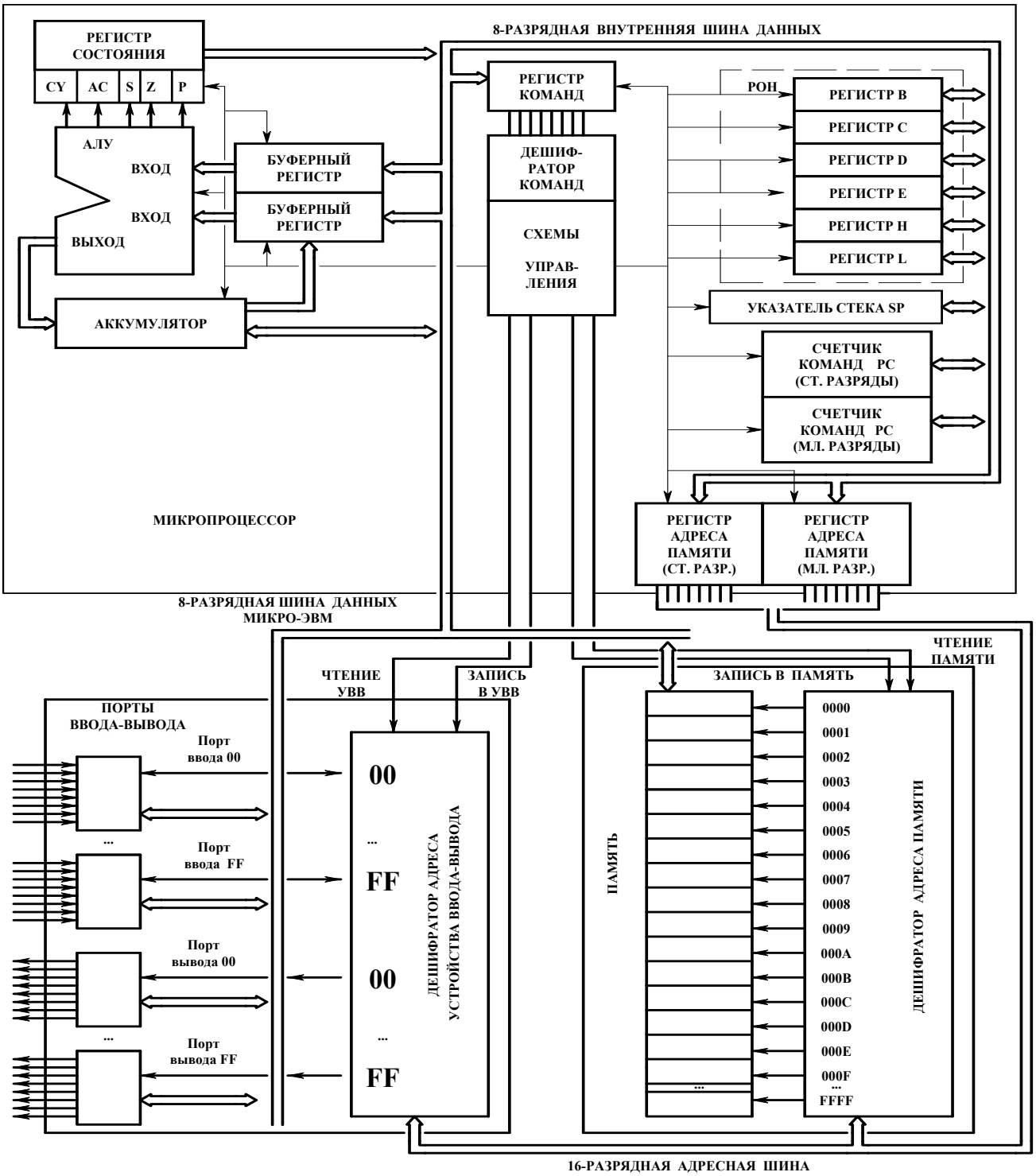


Рис. 7.1. Структурная схема микроЭВМ

Аккумулятор – универсальный регистр микропроцессора при различных манипуляциях с данными. Большинство арифметических и логических операций осуществляется путем использования АЛУ и аккумулятора. Любая из таких операций над двумя словами данных (*операндами)* предполагает размещение одного из них

в аккумуляторе, а другого в памяти или регистре. Так, при сложении двух слов, называемых условно А и В и расположенных в аккумуляторе и памяти соответственно, результирующая сумма С загружается в аккумулятор, замещая слово А. Результат операции АЛУ тоже размещается в аккумуляторе стирая его исходное содержимое.

Операцией другого типа, использующей аккумулятор, является программируемая передача данных из одной части микропроцессора в другую – это пересылка данных между портом ввода-вывода и областью памяти, между двумя областями памяти.

Микропроцессор может выполнять некоторые действия над данными непосредственно в аккумуляторе. Например, аккумулятор может быть очищен путем записи двоичных нулей во все его разряды, установлен в единичное состояние посредством записи двоичных единиц во все разряды. Содержимое аккумулятора можно сдвигать влево или вправо, получать его инвертированное значение, а также выполнять другие операции.

Для выполнения любой операции над данными, поступающими с внутренней шины данных микропроцессора, необходимо поместить их в аккумулятор. В свою очередь аккумулятор может пересылать данные на эту шину.

Счетчик команд предназначен для хранения адреса выполняемой команды. Перед выполнением программы счетчик команд загружается числом – адресом области памяти, содержащей первую команду программы. На рис. 7.1 регистр адреса памяти и адресная шина расположены ниже счетчика команд. Адрес области памяти, содержащей первую команду программы, пересылается из счетчика команд в регистр адреса памяти, после чего содержимое обоих регистров становится одинаковым. После извлечения команды из памяти микропроцессор автоматически дает приращение содержимому счетчика команд. Это приращение счетчик команд получает в тот момент, когда микропроцессор начинает выполнять команду, только что извлеченную из памяти. Следовательно, начиная с этого момента, счетчик команд «указывает», какой будет следующая команда. Счетчик команд содержит адрес следующей выполняемой команды на протяжении всего времени выполнения текущей команды.

Регистр адреса содержит адрес ячейки памяти, к которой обращается процессор для считывания команды или данных. Выход этого регистра является адресной шиной. Регистр адреса подключен к внутренней шине данных МП и он может

загружаться от различных источников. Большинство МП располагают командами, позволяющими загружать этот регистр содержимым счетчика команд, регистров общего назначения или какой-либо области памяти.

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

В течение подцикла выполнения команды содержимое регистра адреса памяти зависит от выполняемой команды. Если в соответствии с командой микропроцессор должен произвести еще одно обращение к памяти, то регистр адреса памяти подлежит вторичному использованию в процессе обработки этой команды.

Регистр команд предназначен для хранения текущей выполняемой команды, эта функция реализуется микропроцессором автоматически с началом цикла выборка-выполнение, называемого также *машинным циклом*.

Сначала команда извлекается из памяти, затем счетчик команд настраивается на указание следующей команды, подлежащей выполнению. При извлечении команды из соответствующей области памяти копия команды помещается на внутреннюю шину данных и пересылается в регистр команд. После этого начинается выполнение команды. Дешифратор команд считывает содержимое регистра команд (КОП) и настраивает УУ на заданную операцию.

Буферные регистры предназначены для временного хранения одного слова данных. Один из этих регистров, ближайший к аккумулятору (рис. 7.1), называется *буфером аккумулятора*. В другой буферный регистр на временное хранение поступают данные с внутренней шины микропроцессора. Необходимость в таком регистре вызвана отсутствием в АЛУ своего запоминающего устройства. В состав АЛУ включены только комбинационные схемы, и поэтому при поступлении исходных данных на входе АЛУ немедленно появляются результирующие данные на его выходе как следствие выполнения операций данной программы.

Если на вход описанного выше буферного регистра могут поступать данные только с внутренней шины данных микропроцессора, то на вход другого буферного регистра, именуемого буфером аккумулятора, данные могут поступать, кроме того, и с выхода аккумулятора. Буфер аккумулятора позволяет избежать ситуации, при которой вход и выход АЛУ одновременно подсоединены к одной и той же точке схемы.

Регистр состояний предназначен для хранения результатов проверок, осуществляемых в процессе выполнения программы. Разряды регистра состояния принимают значение «1» или «0» в зависимости от выполняемой операции. В состав регистра входят флаги: S – знак; CY – перенос; AC – вспомогательный перенос; P – четность; Z – ноль.

Шина данных предназначена для передачи данных. По *линиям данных* в любой определенный момент времени передается одно слово данных в микропроцессор или в обратном направлении. В отличие от адресных шин, шина данных является двунаправленной и по ней передаются данные с внутренней шины микропроцессора в память или блок ввода-вывода либо из памяти или блока ввода-вывода на эту шину.

Шина управления предназначена для управления регистрами МП и контроля за их работой. С помощью управляющих сигналов УУ сообщает, когда передавать (получать) информацию на шину данных.

Шина адреса используется для идентификации областей памяти или устройств, с которыми микропроцессору необходима связь. Адресные линии передают данные только из микропроцессора, в обратном направлении передача запрещена.

Память микроЭВМ состоит из ячеек, имеющих индивидуальный адрес. Каждая ячейка памяти имеет свой номер – адрес его местоположения. Адрес используется всякий раз, когда необходимо записать данные в память или прочесть данные из памяти. Адрес (двоичное число) появляется на адресной шине микропроцессора и декодируется дешифратором адреса памяти. Если на адресную шину помещен адрес допустимого значения, то схемы управления микропроцессора вырабатывают и посылают по линиям управления импульсные сигналы «Чтение памяти» или «Запись в память». Эти сигналы информируют микропроцессор о необходимости подачи данных с шины для записи в выбранную область памяти. При записи данных в память предыдущее содержимое соответствующей ее области стирается и заменяется записываемой информацией. Чтение данных из памяти не меняет содержимое области, в которой они находятся. Одни и те же данные можно считывать любое количество раз.

Блок ввода-вывода микроЭВМ состоит из портов ввода и вывода. Микропроцессор позволяет подключать к шинам адресов до 256 портов ввода и до 256 портов вывода. Порты 00 или 01 являются портами входными. Каждый порт имеет восемь входных линий. При адресации порта 00 или 01 данные, поступающие по этим

линиям, подаются на шину данных микроЭВМ, затем микропроцессор помещает эти данные в аккумулятор. Порты 02 и 03 являются выходными: каждый из них имеет восемь выходных линий. Состояние каждой выходной линии хранится в буфере данных. При обращении к порту 02 или 03 восемь бит данных, имеющихся на шине данных микроЭВМ, записываются в буфер данных адресуемого выходного порта. Эти данные буфер хранит для возможного их использования устройствами, внешними по отношению к микроЭВМ. Данные сохраняются до следующего обращения к этому порту или до тех пор, пока не выключат питание микроЭВМ.

Порты 00, 01, 02 и 03 называются портами параллельного ввода-вывода, поскольку данные пересылаются по восьми параллельным линиям. Принцип такого «параллелизма» заложен в основу организации системы ввода-вывода. Последовательный ввод-вывод применяется, когда возникает необходимость передавать данные на большие расстояния. В пункте приема устанавливается преобразователь последовательного кода в параллельный, подаваемый по восьми параллельным линиям.

Дешифратор адреса блока ввода-вывода подобен дешифратору адреса памяти. Поскольку имеется только четыре порта ввода-вывода, достаточно располагать двумя адресными линиями. Дешифратор адреса портов ввода-вывода декодирует все адреса, пересылаемые микропроцессором. Порты ввода-вывода отвечают на запрос, передаваемый в форме их адресов, только тогда, когда имеет место совпадение импульсов «Чтение ввода-вывода» или «Записи ввода-вывода» с декодированием этих адресов.

Шина микроЭВМ предназначена для передачи нескольких видов сигналов. Она состоит из восьми двунаправленных линий данных, т. е. по этим линиям можно передавать данные как в микропроцессор, так и из него. Эту шину можно рассматривать как продолжение внутренней шины данных микропроцессора.

Шестнадцать адресных линий адреса используются для адресации как памяти так и устройств ввода-вывода. Кроме того, по адресной шине передаются сигналы питания, некоторые сигналы синхронизации и управления, а также с помощью этой шины осуществляется заземление всех блоков.

Структура микроЭВМ, рассмотренная в п. 7.1, ориентирована на микропроцессор КР580ИК80А, предназначенный для обработки 8-разрядных двоичных чисел.

7.2. Система команд и способы адресации

Программа состоит из отдельных команд. Каждая команда предписывает определенное действие (например, сложить два числа). Для реализации данного действия команда разбивается на отдельные операции (чтение команды, формирование адреса операндов, выполнении заданного действия). *Система команд* — это совокупность правил и форм представления управляющей информации для записи программ, пригодных для непосредственного выполнения техническими средствами ЭВМ.

Система команд характеризуется:

- структурой и форматом команды;
- списком выполняемых операций (кодов команд);
- количеством адресов в команде (адресностью);
- способом адресации.

Структура команды — это схема расположения информации в команде.

Формат команды – структура команды с разметкой номеров разрядов, определяющей границы ее полей.

Команды, как и любая информация в ЭВМ, кодируются двоичными словами, которые должны содержать в себе следующие виды информации (рис. 7.2):

- тип операции (код), которую следует реализовать в данной команде (КОП);
- место в памяти, откуда следует взять первый операнд (А1);
- место в памяти, откуда следует взять второй операнд (А2);
- место в памяти, куда следует поместить результат (А3).

КОП	А1	А2	А3

Рис. 7.2. Формат команды

Список выполняемых операций – это перечень всех принятых в данной ЭВМ командных операций и их кодировка в операционном поле команды.

Адресность ЭВМ определяется количеством адресов, указанных в команде.

Способ адресации показывает, каким образом по информации, содержащейся в поле адреса команды, получается исполнительный адрес.

Различные МП имеют различный набор команд, способы адресации в них одинаковы.

Способом адресации называется тип обращения к данным.

Основные способы адресации:

- неявная адресация;
- непосредственная адресация;
- прямая адресация;
- косвенная адресация.

Неявная адресация – это способ адресации данных (рис. 7.3), когда в составе команды указываются:

- КОП;
- адрес источника данных;
- адрес приемника данных.

Адрес приемника и источника указаны в команде не явно, говорят, что они «встроены» в команду. При этом способе используются однобайтные команды.

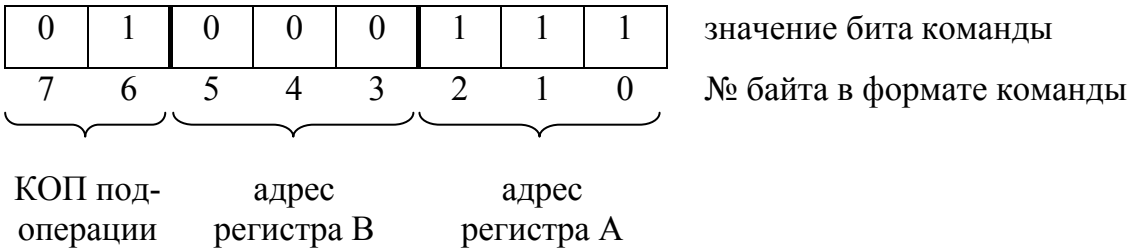


Рис. 7.3. Неявная адресация

Команды с неявной адресацией отличаются наибольшей величиной быстродействия, так как для ее выполнения затрачиваются два микроцикла: выборка и операция выполнения.

Непосредственная адресация – способ определения операнда (рис. 7.4), который содержит в адресной части сам операнд, а не адрес операнда. Такой операнд называется непосредственным операндом, поскольку он автоматически вызывается из памяти одновременно с командой, следовательно, он сразу непосредственно становится доступным.

Команда с непосредственным адресом для загрузки константы 4 в регистр 1.

MVI	R1	4

Рис. 7.4. Непосредственная адресация

При непосредственной адресации не требуется дополнительного обращения к памяти для вызова операнда. Однако у такого способа адресации есть и некоторые недостатки. Во-первых, таким способом можно работать только с константами. Во-вторых, число значений ограничено размером поля. Тем не менее эта технология используется во многих архитектурах для определения целочисленных констант.

Прямая адресация способ определения операнда – дать его полный адрес. Такой способ называется прямой адресацией. Как и непосредственная адресация, прямая адресация имеет некоторые ограничения: команда всегда будет иметь доступ только к одному и тому же адресу памяти. То есть значение может меняться, а адрес – нет.

Команды с прямой адресацией могут иметь длину, равную 2 или 3 байтам.

1 байт – КОП;

1–2 байта – для адреса.

Отличие прямой и непосредственной адресации: при непосредственной адресации данные располагаются в структуре команды и доступ из других команд к ним затруднен. При прямой адресации данные располагаются в специально выделенном по указанному адресу участку памяти, что обеспечивает доступ к ним, другим командам.

При прямой адресации используются микропрограммы:

- выборка кода операции;
- декодирование;
- выборка двух байт адреса;
- два цикла обращения к памяти;
- выполнение операции.

Косвенная адресация – этот способ адресации реализуется командой, содержащей код операции и номер регистра, который содержит адрес, где располагаются обрабатываемые данные в памяти.

При таком способе адресации определяемый операнд берется из памяти или отправляется в память, но адрес не зафиксирован жестко в команде, как при прямой адресации. Вместо этого адрес содержится в регистре. Если адрес используется таким образом, он называется указателем. Преимущество косвенной адресации состоит в том, что можно обращаться к памяти, не имея в команде полного адреса.

Таким образом, при косвенной адресации используются двухбайтовые команды: первый байт для кода операции, второй байт указывает на регистровую пару, которая содержит адрес ячейки памяти, в которой записан байт данных.

7.3. Типы команд микроЭВМ

Большинство команд осуществляет пересылку или обработку данных, расположенных в памяти или в одном из регистров микропроцессора. Для выполнения команда передается в регистр команд, дешифратор и схемы управления, где она идентифицируется, в результате чего формируются сигналы, направляемые в другие части микропроцессора. С помощью этих сигналов выполняются операции, предписываемые командой.

Команды в микроЭВМ подразделяются:

- команды перемещения (пересылки данных);
- арифметические;
- логические;
- команды ветвления (перехода).

Команды пересылки данных

Команды *пересылки данных,* называемые также *командами перемещения* данных, служат для пересылки данных в различные устройства хранения информации, которыми располагает микроЭВМ, а также для пересылки данных из этих устройств. К числу областей хранения информации относятся как ячейки памяти, так и регистры. В зависимости от того, какие устройства микроЭВМ участвуют в пересылке данных, различают команды: загрузка в регистр (MVI), пересылка данных из регистров (MOV), запись в память (STA) и пересылка данных из памяти (LDA). Как и любая другая команда, команда пересылки данных состоит из двух частей: кода операции и

адресной части. Во всех командах пересылки данных должны быть указаны как источник, так и место назначения данных. Код операции в команде пересылки данных задает источник данных и способ адресации. Адресация может быть одного из трех видов: непосредственная, прямая и косвенная.

При использовании команды *пересылки из регистра в регистр* (MOV) в регистр r1 загружается копия данных, содержащихся в регистре r2. Так, при выполнении команды MOV B, A копия данных, находящихся в аккумуляторе (регистре A), записывается в регистр B. Это 1-байтовая команда, реализуемая за два микроцикла. Если результат выполнения команды содержит 1 в старшем разряде или равен 0, то устанавливается в 1 соответствующий разряд регистра состояния.

При использовании команды *запись аккумулятора в память прямая* (STA) данные из аккумулятора записываются в память. Второй и третий байты команды указывают на область памяти, в которую производится запись данных. Например, по команде STA 001F содержимое аккумулятора записывается в область памяти 001F. Это 3-байтовая команда, реализация которой занимает пять микроциклов. Значение разрядов регистра состояния в результате выполнения операции не изменяется.

Арифметические и логические команды

По команде с*ложение с памятью прямое (ADD)* содержимое области памяти, адрес которой указан в команде, прибавляется к содержимому аккумулятора (регистра A). Второй и третий байты команды составляют адрес области памяти, в которой находится одно из слагаемых. Результат (сумма) помещается в аккумулятор, исходное содержимое стирается. Это 3-байтовая команда, для выполнения которой затрачиваются четыре микроцикла процессора. Если результат выполнения команды содержит 1 в старшем разряде или равен нулю, либо если возникает перенос из старшего разряда результата, то соответствующие разряды регистра состояния устанавливаются в 1.

Команда *сложение с переносом (ADC)* дает возможность производить сложение двух чисел вместе с переносом, полученным в предыдущей операции. При каждом выполнении команды *ADC* соответствующим образом устанавливаются в регистре состояния разряды, указывающие на равенство результата нулю, наличие у него знака минус и наличие переноса из старшего разряда.

Поэтому исходное содержимое и аккумулятора, и регистра состояния утрачивается, как и при выполнении других команд.

Команда *сравнения (CMP)* производит сравнение операнда, находящегося в одном из регистров с содержимым аккумулятора. Результат операции *сравнение* не отражается на содержимом аккумулятора, а лишь изменяет установку регистра состояний (1 или 0).

Команды перехода или ветвления

Команда перехода (JMP), называемая также *командой ветвления*, позволяет организовать в программах циклы и разветвления. При использовании команды перехода изменяется содержимое счетчика команд. Содержимое второго и третьего байтов команды перехода пересылается в счетчик команд во время фазы выполнения. Тогда при очередном цикле выборки микропроцессор извлекает команду из области памяти, на которую указывают второй и третий байты команды перехода. Таким образом, происходит переход по указанному в команде адресу программы и выполняются (повторяются) команды новой последовательности. Это продолжается до тех пор, пока не будет опять выполнена команда перехода или произведено выключение микроЭВМ.

7.4. Выполнение команды микропроцессором

МикроЭВМ работает в форме последовательного выполнения машинных команд, хранящихся в ячейках памяти. Время выполнения каждой команды состоит из двух интервалов – выборки и собственно выполнения. Каждый из обоих интервалов занимает во времени некоторое число циклов генератора тактовой частоты, зависящее от типа команды (рис. 7.5). В свою очередь, каждый из циклов генератора тактовой частоты соответствует выполнению одного элементарного шага команды, называемого *микрооперацией.*

В табл. 7.1 представлены шаги (микрооперации) выполнения команды микропроцессором – чтение числа из заданной ячейки памяти, прибавление к нему единицы и сохранение результата в аккумуляторе.

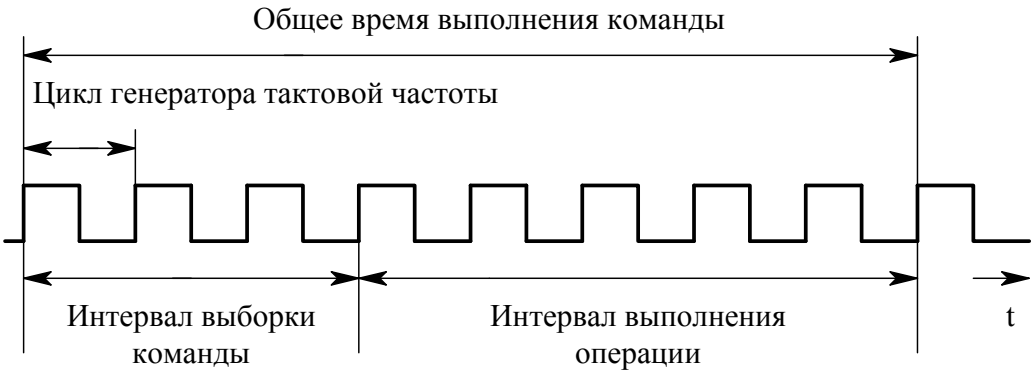


Рис. 7.5. Интервалы выполнения команды

Таблица 7.1

Наименование интервала	Цикл генератора тактовой частоты	Выполняемая микрооперация	Активные линии магистрали управления
Выборка команды из памяти	1	Пересылка адреса команды из программного счетчика по МА в регистр адреса памяти	2 и 5
	2	Пересылка кода адресуемой команды из ячейки памяти по МД в регистр команд	3 и 6
		Наращивание содержимого программного счетчика (формирование адреса следующей команды)	1
Выполнение операции	3	Пересылка адреса операнда из регистра команд по МА в регистр адреса памяти	4 и 5
	4	Пересылка адресуемого операнда из ячейки памяти по МД в операционный блок	6 и 8
	5	Выполнение заданной операции над операндом в операционном блоке	9
	6	Запись результата операции в аккумулятор	12

В *первом цикле* адрес команды, хранящийся в программном счетчике, пересылается по МА в регистр адреса памяти. С этой целью УУ выдает одновременно два управляющих сигнала по линиям 2 и 5 (рис. 7.6).

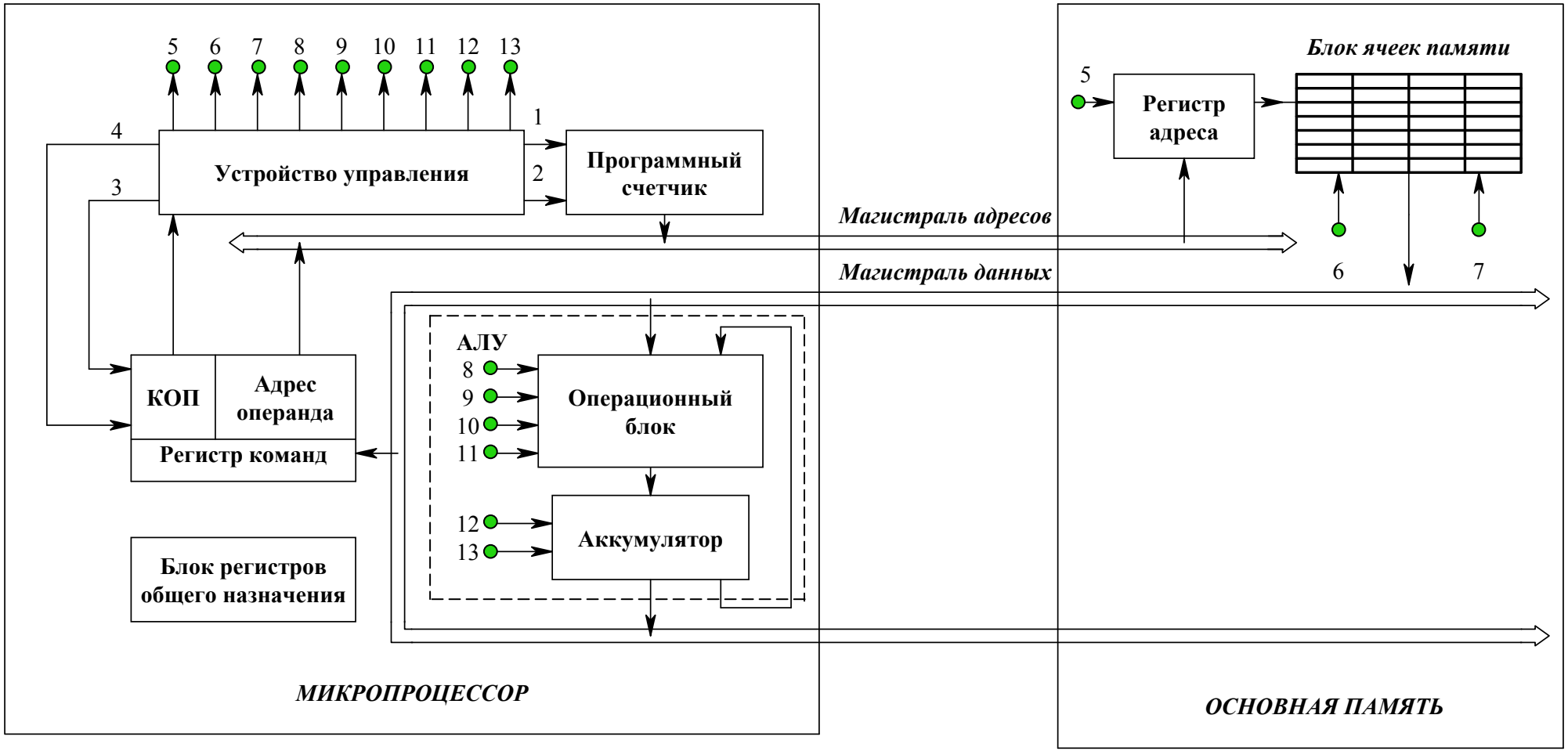


Рис. 7.6. Структурная схема МП

Во *втором цикле,* путем одновременной активизации линий 3 и 6, код команды, адрес которой теперь имеется в регистре адреса памяти, пересылается из адресуемой ячейки памяти по МД в регистр команд. В этом же цикле путем активизации линии 1 от УУ одновременно наращивается содержимое программного счетчика так, чтобы получился адрес следующей команды программы. Обычно этот адрес получается прибавлением одной и той же константы (например, 1) к адресу текущей команды. Эта константа формируется автоматически.

Второй цикл завершает интервал выборки команды из памяти в микропроцессор. Код команды находится в регистре команд. Одна часть этого кода содержит КОП, а другая часть — адрес операнда (числа) в памяти, над которым должна выполняться требуемая операция. Далее начинается интервал выполнения операции, заложенной в данной команде. Конкретная последовательность выдачи управляющих сигналов от УУ на интервале выполнения операции полностью зависит от КОП. В рассматриваемом примере интервал выполнения операции содержит циклы 3, 4, 5 и 6.

В *третьем цикле* адрес операнда из регистра команд пересылается по МА в регистр адреса памяти подачей сигналов от УУ по линиям 4 и 5.

В *четвертом цикле* адресуемый операнд пересылается из соответствующей ячейки памяти по МД в операционный блок путем активизации линий 6 и 8.

В *пятом цикле* над операндом, засланным в операционный блок, осуществляется требуемая операция прибавления единицы. Увеличение содержимого операционного блока на единицу достигается самим операционным блоком при подаче на него управляющего сигнала по линии 9 от УУ.

В *шестом цикле* увеличенное на единицу содержимое операционного блока пересылается в аккумулятор под действием управляющего сигнала по линии 12 от УУ. На этом выполнение команды завершается.

Так как в этот момент программный счетчик уже содержит адрес следующей команды, то в очередном цикле генератора тактовой частоты немедленно начинается первый шаг новой команды. Эта новая команда может, например, осуществить суммирование числа в аккумуляторе (т. е. результата предыдущей команды) с числом из какой-нибудь ячейки памяти и записать полученную сумму в другую ячейку памяти или же в один из регистров общего назначения. Выполняя одну команду за другой, МП продолжает работать до тех пор, пока не встретится команда останова или оператор принудительно не остановит микроЭВМ нажатием соответствующей клавиши.

7.5. Организация интерфейса в микроЭВМ

Интерфейс (interface) – совокупность средств сопряжения и связи, обеспечивающая эффективное взаимодействие систем или их частей. В интерфейсе предусмотрены вопросы сопряжения на механическом (число проводов, элементы связи, типы соединений, разъемы, номера контактов и т. п.) и логическом (сигналы, их длительность, полярность, частоты и амплитуда, протоколы взаимодействия) уровнях. В современных интерфейсах для формирования стандарта подключения устройств к системе широко используются наборы микросхем, генерирующих стандартные сигналы.

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

В *функции интерфейса* входят дешифрация адреса устройств, синхронизация обмена информацией, согласование форматов слов, дешифрация кода команды, связанной с обращением к периферийному устройству, электрическое согласование сигналов и другие операции. Интерфейс должен осуществлять совместимость между всеми компонентами микроЭВМ по быстродействию, форме представления обмениваемой информации, архитектуре центрального процессора и электрическим характеристикам.

Из всего многообразия типов микроЭВМ можно выделить две основные организации интерфейса между процессором, памятью и периферийными устройствами: двухшинную, или интерфейс с изолированной системой шин, и одношинную, или интерфейс с общей шиной.

Интерфейс с изолированной шиной. Особенностью этого интерфейса является раздельная адресация памяти и периферийных устройств при обмене информацией путем использования отдельных групп команд для обмена информацией с периферийными устройствами и памятью. Обмен с периферийными устройствами производится с помощью двухбайтных команд ввода IN PORT и вывода OUT PORT, а управление обменом — под воздействием управляющих сигналов ввода и вывода,

формируемых системным контроллером при выполнении соответствующих команд. Адрес из МП передается по МА, передача данных выполняется по МД между аккумулятором и буферным регистром интерфейсного модуля (портом) периферийного устройства. Обмен информацией между МП и памятью производится по командам передачи данных.

Основным недостатком организации интерфейса с изолированной шиной является осуществление обмена обязательно через аккумулятор МП, что, как правило, требует выполнения четырех команд.

Интерфейс с общей шиной. При данной организации интерфейса все адресное пространство делится между периферийными устройствами и ячейками памяти, причем обращение к периферийным устройствам осуществляется посредством набора команд, используемых для передачи данных с памятью. Достоинствами данного интерфейса являются:

– возможность расширения набора команд для обращения к периферийным устройствам, что позволяет сократить объем программ и повысить быстродействие;

– возможность обмена информацией между любыми регистрами МП;

– возможность внепроцессорного обмена данными между ячейками памяти.

Недостатком интерфейса с общей шиной является усложнение дешифрирующих схем и сокращение области памяти.

Общая шина (ОШ) это — магистраль, которая связывает между собой все периферийные устройства, запоминающие устройства и процессор. Все устройства, включая процессор, подсоединены в *общую шину* параллельно.

Взаимодействие устройств на ОШ основано на принципе *ведомый-ведущий (задатчик — исполнитель).* Задатчик получает ОШ в свое распоряжение на промежуток времени для определения операции ввода-вывода с исполнителем. Одно и то же устройство может быть и задатчиком, и исполнителем (в зависимости от типа операции). На ОШ можно организовать автономный обмен данными непосредственно между двумя ВУ без привлечения для этого ресурсов процессора и ОП.

Все передачи по ОШ осуществляются асинхронно по *методу запрос-ответ.* В случае неполучения ответного сигнала от исполнителя задатчик фиксирует ошибку в работе ОШ и выполняет операцию прерывания.

В структуре связей микроЭВМ разделяют *внутренний интерфейс,* объединяющий БИС процессора, модули ОЗУ, ПЗУ и схемы управления вводом-выводом, и *внешний интерфейс,* обеспечивающий сопряжение между внутренней магистралью и периферийными устройствами.

Таким образом, наиболее распространенными интерфейсами микроЭВМ являются интерфейсы ***с общей шиной***, представляющие собой совокупность совместимых и частично совместимых магистралей.

Контрольные вопросы

1. Назовите состав структурной схемы микроЭВМ.
2. Назначение основных регистров микроЭВМ.
3. Чем характеризуется система команд?
4. Способы адресации в микроЭВМ.
5. Основные типы команд в микроЭВМ.
6. Назначение команды ADD.
7. Какой из регистров изменяет свое состояние при команде сравнения?
8. Какие основные типы интерфейсов используются в микроЭВМ?

8. АРХИТЕКТУРА ПЭВМ

8.1. Системный блок ПЭВМ

Устройства, находящиеся внутри системного блока, называют *внутренними,* а устройства, подключаемые к нему, – *внешними.* Внешние дополнительные устройства, предназначенные для ввода, вывода и длительного хранения данных, также называют *периферийными.*

Конструктивно системный блок состоит из корпуса, в котором установлены системная (материнская) плата, накопители на гибких и жестких магнитных дисках (НГМД, НЖМД), накопители на оптических дисках (НОД); видеокарта; звуковая карта; сетевая карта и другие устройства (рис. 8.1).

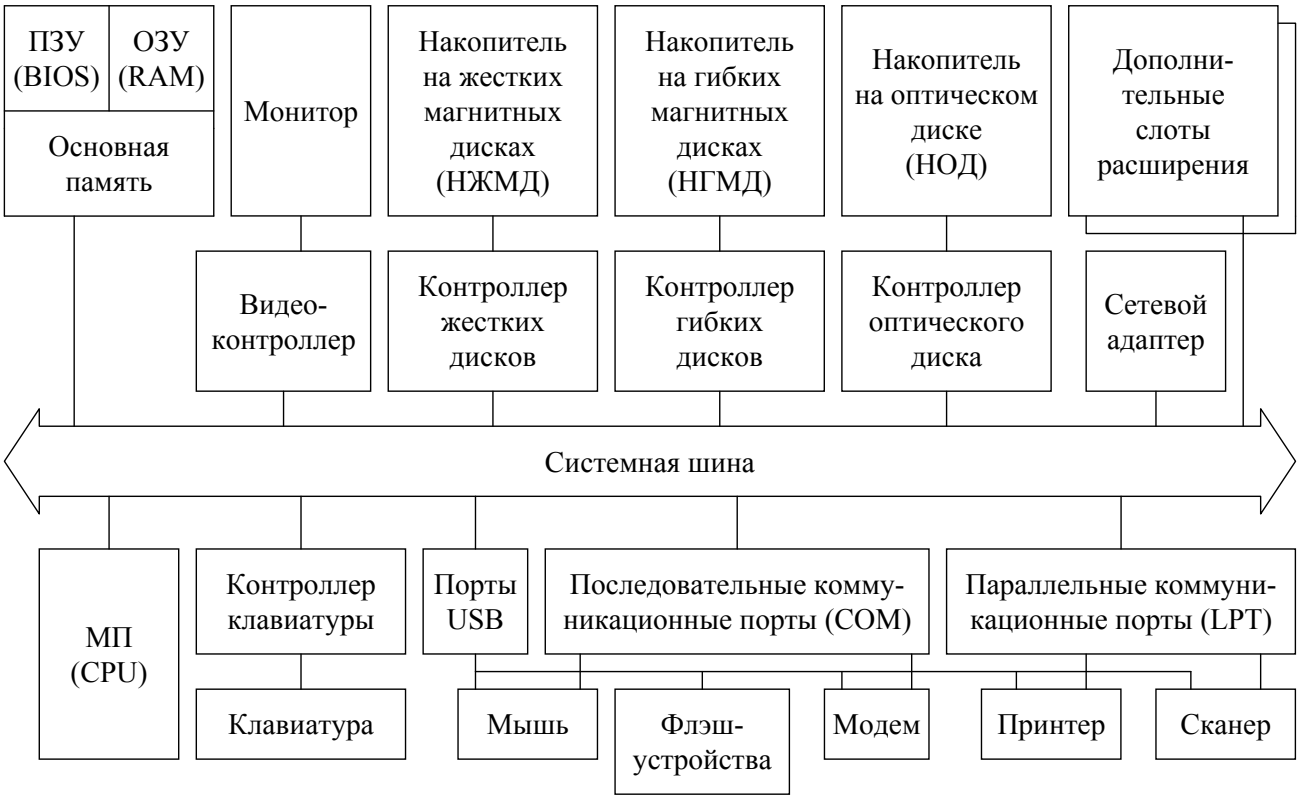


Рис. 8.1. Системный блок ПЭВМ

Корпуса персональных компьютеров выпускают в горизонтальном (*desktop)* и вертикальном (*tower)* исполнении. Корпуса, имеющие вертикальное исполнение, различают по габаритам: *полноразмерный (big-tower), среднеразмерный (midi-tower)* и

малоразмерный (mini-tower). Корпуса персональных компьютеров поставляются вместе с блоком питания и, таким образом, мощность блока питания также является одним из параметров корпуса. Мощность блока питания составляет 250–350 Вт и более.

Системная плата (System board) – является основой системного блока, определяющая архитектуру и производительность ПЭВМ. Архитектура системного блока представлена на рис. 8.2.

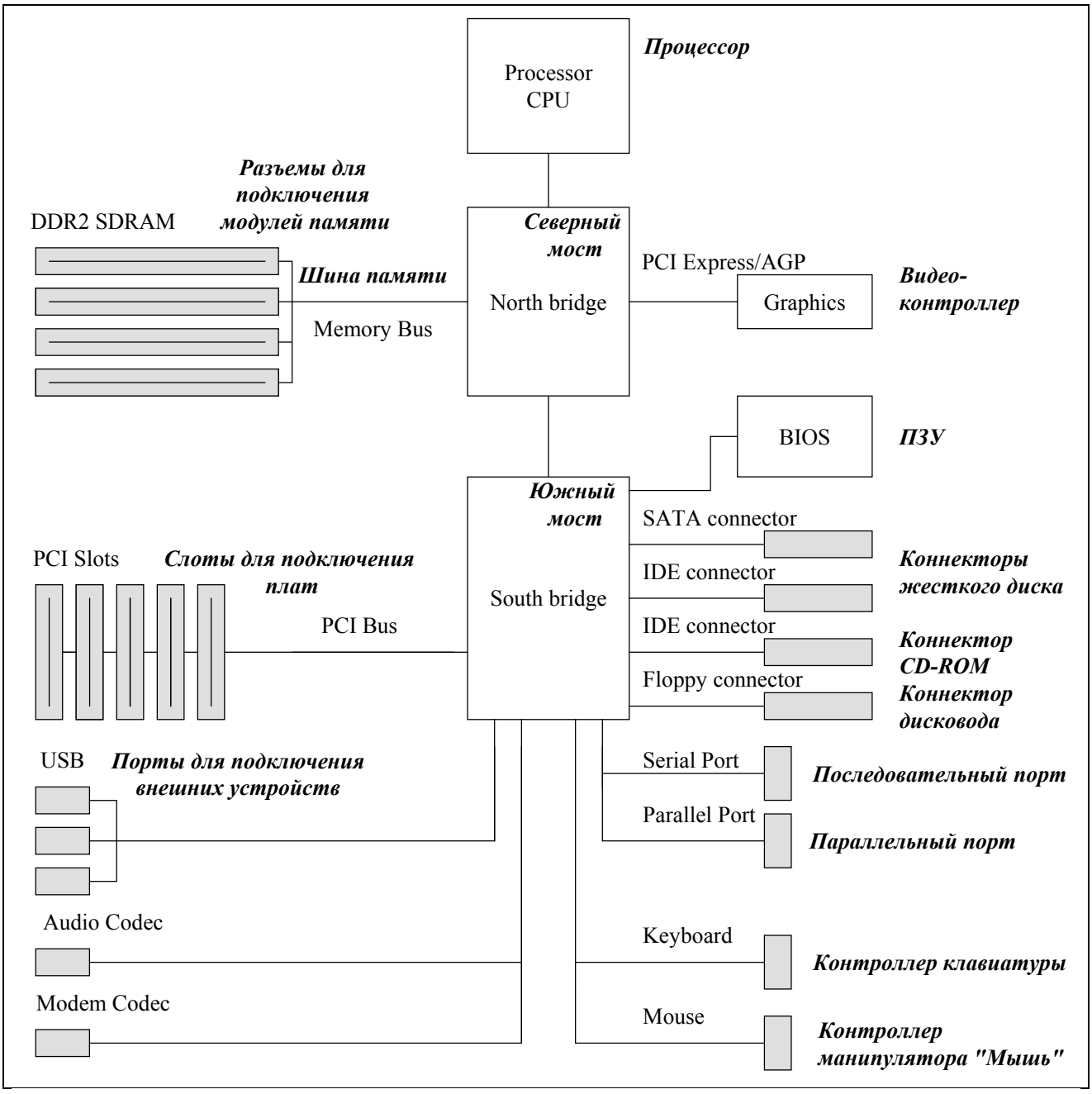


Рис. 8.2. Архитектура системной платы

Устройства, непосредственно осуществляющие процесс обработки информации (вычисления), в том числе микропроцессор, оперативная память и шина, размещаются на системной (материнской) плате компьютера, на ней же располагается и контроллер клавиатуры. 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. Иногда эти контроллеры могут располагаться на системной плате. Наборы микросхем, на основе которых исполняются системные платы, называют *чипсетами* (ChipSets).

На системной плате размещаются:

– *микропроцессор (CPU – Central Processor Unit)* – основная микросхема, выполняющая арифметические и логические операции, заданные программой, управляет вычислительным процессом и координирует работу всех устройств компьютера;

– *микропроцессорный комплект (чипсет)* – набор микросхем, управляющих работой внутренних устройств компьютера и определяющих основные функциональные возможности материнской платы. В настоящее время большинство чипсетов материнских плат выпускаются на базе двух микросхем, получивших название «северный мост» и «южный мост» (рис. 8.2). «Северный мост» управляет взаимосвязью четырех устройств: процессора, оперативной памяти, шины *PCI Express* или AGP. «Южный мост» называют также функциональным контроллером. Он выполняет функции контроллера жестких и гибких дисков, функции контроллера клавиатуры, мыши, шины *USB* и т. п.;

– *память* (внутренняя – системная, включающая ОЗУ и ПЗУ, и внешняя дисковая). *ПЗУ (ROM, Read Only Memory* – память только для чтения) служит для хранения неизменяемой (постоянной) программной и справочной информации. *ОЗУ* (*RAM, Random Access Memory* – память с произвольным доступом) предназначено для оперативной записи, хранения и считывания информации (программ и данных), непосредственно участвующей в информационно-вычислительном процессе, выполняемом ПК в текущий период времени. Дисковая память относится к внешним устройствам ПК и используется для долговременного хранения любой информации, которая может когда-либо потребоваться для решения задач, в ней, в частности, хранится все

программное обеспечение компьютера. В качестве устройств внешней памяти, размещаемых в системном блоке, используются накопители НЖМД, НГМД, НОД и др;

– *контроллеры (адаптеры)* служат для подключения периферийных (внешних по отношению к процессору) устройств к шинам микропроцессора, обеспечивая совместимость их интерфейсов. Они осуществляют непосредственное управление периферийными устройствами по запросам микропроцессора. Контроллеры реализуются, как правило, на отдельных печатных платах, часто называемых адаптерами устройств (от лат. *adapto* – преобразовываю);

– *системная шина* – основная интерфейсная система компьютера, обеспечивающая сопряжение и связь всех его устройств между собой. Системная шина включает: *шину данных*, содержащую провода и схемы сопряжения для параллельной передачи всех разрядов числового кода (машинного слова) операнда, *шину адреса*, состоящую из проводов и схем сопряжения для параллельной передачи всех разрядов кода адреса ячейки основной памяти или порта ввода-вывода внешнего устройства, *шину управления*, содержащую провода и схемы сопряжения для передачи инструкций (управляющих сигналов) во все блоки компьютера, и *шину питания,* имеющую провода и схемы сопряжения для подключения блоков ПК к системе энергопитания. Системная шина обеспечивает три направления передачи информации: между микропроцессором и внутренней (основной) памятью, между микропроцессором и портами ввода-вывода внешних устройств, между внутренней (основной) памятью и портами ввода-вывода внешних устройств (в режиме прямого доступа к памяти).

На рис. 8.3 представлена материнская плата чипсета Intel Q 965 с типом разъема Socket 775 процессора Pentium 4, двухядерного, частотой шины 533, 800, 1066 МГц, оперативной памятью DDR2 (800/667/533 МГц), шиной PCI Express и сетевой картой передачи данных до 1000 Мбит/с.

Накопитель на гибких магнитных дисках (FDD – Floppy Disk Drive) – предназначен для записи (считывания) информации на дискеты. Основная характеристика: емкость.

Накопитель на жестком магнитном диске (HDD – Hard Disk Drive) – предназначены для долговременного хранения информации. Обеспечивают наиболее быстрый среди накопителей доступ к данным. Основные характеристики: емкость измеряемая в ГБайтах (1–200 и более), быстродействие (3600, 7200, 10000).

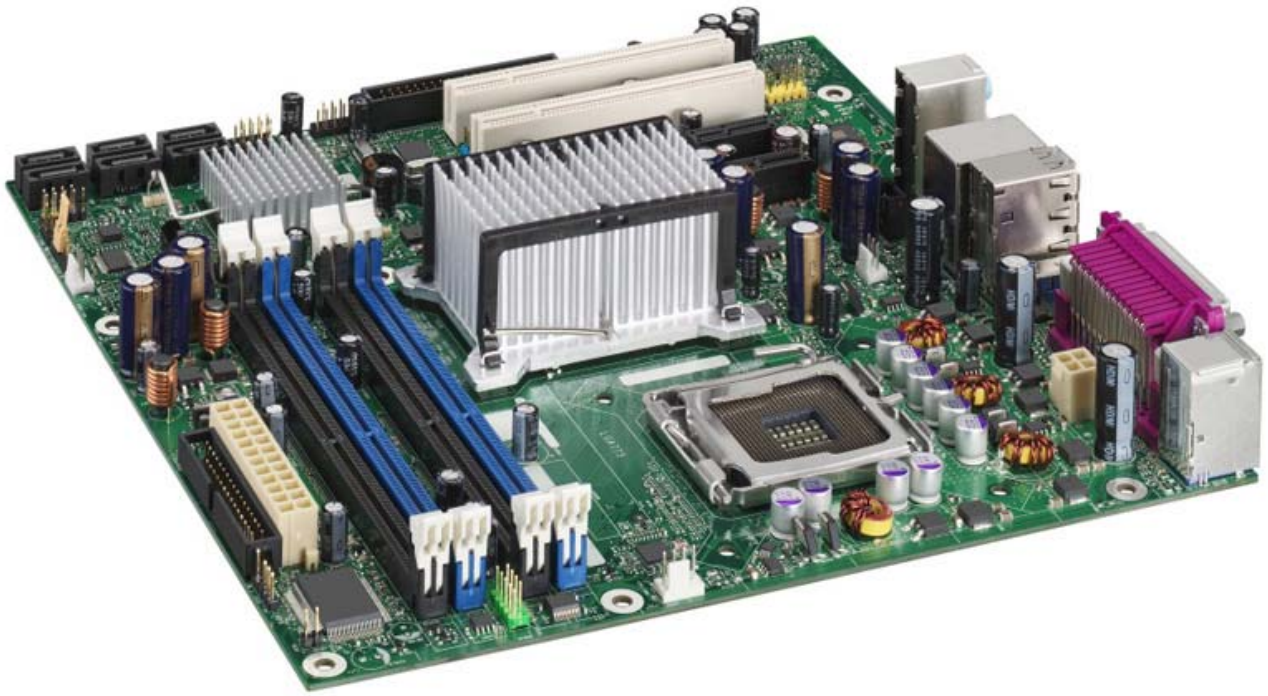


Рис. 8.3. Материнская плата чипсета Intel Q 965

Накопитель на оптических дисках (CD-ROM-Compact Disk Read Only Memory) – предназначен для чтения оптических дисков, для хранения информации используется изменение оптических свойств (степени отражения) поверхности носителя. Основная характеристика – скорость считывания.

Видеокарта (Video Card) – видеоадаптер, обеспечивающий формирование видеосигнала и определяющий изображение на мониторе. Основная характеристика видеоадаптера: объем видеопамяти измеряемый в Мегабайтах.

Видеоускорение — одно из свойств видеоадаптера, которое заключается в том, что часть операций по построению изображений может происходить без выполнения математических вычислений в основном процессоре компьютера, а чисто аппаратным путем — преобразованием данных в микросхемах *видеоускорителя.* Видеоускорители могут входить в состав видеоадаптера (в таких случаях говорят о том, что видеокарта обладает функциями аппаратного ускорения), но могут поставляться в виде отдельной платы, устанавливаемой на материнской плате и подключаемой к видеоадаптеру.

Звуковая карта (Sound Card) – служит для воспроизведения звука. В своем составе имеет цифровой канал записи-воспроизведения моно- и стереофонического сигнала, микшер, синтезатор и MIDI-порт. Звуковая карта имеет набор разъемов для подключения внешних аналоговых сигналов (входные – микрофон, линейный вход,

CD-ROM; выходные – линейный выход, выход на колонки или наушники), некоторые платы имеют и цифровой вход от CD-ROM.

Сетевая карта (Net Card) – служит для передачи данных в вычислительных сетях. Различаются по поддерживаемым протоколам доступа к среде передачи данных (Ethernet, Fast Ethernet).

В системном блоке располагается также ***блок питания***, преобразующий переменное напряжение электросети в постоянное напряжение различной полярности и величины, необходимое для питания системной платы и других устройств компьютера, размещенных в системном блоке. Блок питания содержит вентилятор, создающий циркулирующие потоки воздуха для охлаждения системного блока, сетевого энергопитания ПК. Кроме сетевого, в компьютере имеется также автономный источник питания — аккумулятор. К аккумулятору подключается *таймер* – внутримашинные электронные часы, обеспечивающие при необходимости автоматический съем текущего момента времени (год, месяц, часы, минуты, секунды и доли секунд). Таймер продолжает работать и при отключении компьютера от сети.

Важнейшую роль в работе ПЭВМ играет контроллер прерываний.

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

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

8.2. Основные характеристики микропроцессоров, используемых в ПЭВМ

Микропроцессор, или Central Processing Unit (CPU) – функционально-законченное программно управляемое устройство обработки информации, выполненное в виде одной или нескольких БИС или СБИС интегральных схем.

МП выполняет следующие функции:

– вычисление адресов команд и операндов;
– выборку и дешифрацию команд из ОП;
– выборку данных из ОП;
– прием и обработку запросов и команд от адаптеров на обслуживание ВУ;
– обработку данных и их запись в ОП;
– выработку управляющих сигналов для всех узлов и блоков ПК;
– переход к следующей команде.

В настоящее время выпускается несколько сотен различных микропроцессоров, но среди микропроцессоров, используемых в ПЭВМ, наиболее популярными являются микропроцессоры семейства x86.

Конструктивно современный МП представляет собой СБИС, реализованную на одном полупроводниковом кристалле — тонкой пластинке кристаллического кремния прямоугольной формы площадью всего несколько квадратных миллиметров. На ней размещены схемы, реализующие все функции процессора. Кристалл-пластинка обычно помещается в пластмассовый или керамический плоский корпус и соединяется золотыми выводами с металлическими штырьками, чтобы его можно было присоединить к системной плате компьютера.

Внешний вид микропроцессора Intel Pentium 4 Core2 Duo представлен на рис. 8.4.

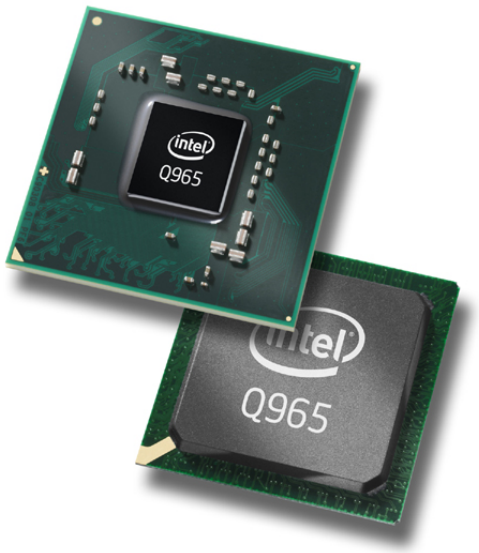


Рис. 8.4. Микропроцессор Intel Pentium 4 Core2 Duo

С внешними устройствами, и, в первую очередь, с оперативной памятью, процессор связан несколькими группами проводников, называемых шинами.

Адресная шина. У большинства современных процессоров адресная шина 32-разрядная, т. е. состоит из 32 параллельных линий. В зависимости от того, есть напряжение на какой-то из линий или нет, говорят, что на этой линии выставлена единица или ноль. Комбинация из 32 нолей и единиц образует 32-разрядный адрес, указывающий на одну из ячеек оперативной памяти. К ней и подключается процессор для копирования данных из ячейки в один из своих регистров.

Шина данных. По этой шине происходит копирование данных из оперативной памяти в регистры процессора и обратно. В компьютерах, собранных на базе процессоров *Intel Pentium,* шина данных 64-разрядная, т. е. состоит из 64 линий, по которым за один раз на обработку поступают сразу 8 байтов.

Шина команд. В большинстве процессоров шина команд 32- и 64-разрядная, хотя существуют 128-разрядные процессоры.

В процессорах семейства x86 различают реальный, защищенный и виртуальный режимы работы.

Реальный режим соответствует возможностям первых процессоров 8086/8088, имеющих 20-разрядную адресную шину, позволяя адресовать не более 1 Мбайт (2^{20}) памяти. Чтобы поддержать совместимость с ранее разработанными программами, все последующие процессоры поддерживают реальный режим, используя при этом свои минимальные возможности.

Защищенный режим появился впервые в МП 80286. В этом режиме, если физическая память полностью загружена, непоместившиеся данные МП располагает на винчестере. При этом он работает не с реальными, а с виртуальными адресами, которые управляются через специальные таблицы, с тем чтобы информацию можно было найти (или снова записать). Эту память называют еще виртуальной памятью, так как фактически она не существует.

Кроме того, в защищенном режиме возможна поддержка мультизадачного режима. При этом *CPU* может выполнять различные программы в выделенные кванты времени, выпадающие на каждую из программ. Начиная с процессора 80386 его архитектура определяет четыре уровня привилегий для защиты кода и данных системы от случайного или преднамеренного изменения со стороны менее привилегированного

кода. Такой метод выполнения кода называют моделью защиты Intel. Уровни привилегий задаются от 0 до 3. Уровень привилегий 0, отведенный под ядро операционной системы, — *режим ядра* максимальный. Уровень привилегий 3, или *режим пользователя,* минимальный.

Виртуальный режим. Впервые начиная с процессора 386 *CPU* могут эмулировать работу нескольких процессоров 8086 и тем самым обеспечить многопользовательский режим таким образом, что на одном ПК могут быть записаны одновременно даже различные операционные системы. Естественно, увеличивается и возможное количество выполняемых приложений.

Основными параметрами МП являются: рабочее напряжение, разрядность, рабочая тактовая частота, коэффициент внутреннего умножения тактовой частоты и размер кэш-памяти.

Рабочее напряжение процессора обеспечивает материнская плата, поэтому различным моделям МП соответствуют разные материнские платы (их надо выбирать совместно). По мере развития МП происходит постепенное понижение рабочего напряжения. Ранние модели процессоров x86 имели рабочее напряжение 5 В. С переходом к процессорам Intel Pentium оно было понижено до 3,3 В, а в настоящее время оно составляет менее 3 В. Причем ядро процессора питается пониженным напряжением 2,2 В и менее. Понижение рабочего напряжения позволяет уменьшить расстояния между структурными элементами в кристалле процессора до десятитысячных долей миллиметра, не опасаясь электрического пробоя. Пропорционально квадрату напряжения уменьшается и тепловыделение в процессоре, а это позволяет увеличивать его производительность без угрозы перегрева.

Разрядность процессора показывает, сколько бит данных он может принять и обработать в своих регистрах за один раз (за один такт). Первые процессоры x86 были 16-разрядными. Начиная с процессора 80386 они имеют 32-разрядную архитектуру. Современные процессоры семейства *Intel Pentium* могут быть 32- и 64-разрядными, и работать с 64-разрядной шиной данных (разрядность процессора определяется разрядностью внутренних регистров).

Рабочая тактовая частота и коэффициент ее внутреннего умножения. В процессоре исполнение каждой команды занимает определенное количество тактов. Тактовые импульсы задает одна из микросхем, входящая в микропроцессорный

комплект (чипсет), расположенный на материнской плате. Чем выше частота тактов, поступающих на процессор, тем больше команд он может исполнить в единицу времени, тем выше его производительность. Первые процессоры x86 могли работать с частотой не выше 4,77 МГц, а сегодня рабочие частоты процессоров уже превосходят несколько миллиардов тактов в секунду (свыше 3 ГГц).

Тактовые сигналы процессор получает от материнской платы, которая по чисто физическим причинам не может работать со столь высокими частотами, как процессор и составляет сотни МГц. Для получения более высоких частот в процессоре происходит внутреннее умножение частоты на коэффициент 3; 3,5; 4; 4,5; 5 и более. Например, у процессора Pentium 4 частота системной шины 4x200 МГц составляет 800 МГц.

Обмен данными внутри процессора происходит в несколько раз быстрее, чем обмен с другими устройствами, например, с оперативной памятью. Для того чтобы уменьшить количество обращений к оперативной памяти, внутри процессора создают буферную область — так называемую *кэш-память.* Это как бы сверхоперативная память. Когда процессору нужны данные, он сначала обращается в кэш-память, и только если там нужных данных нет, происходит его обращение в оперативную память. Принимая блок данных из оперативной памяти, процессор заносит его одновременно и в кэш-память. Высокопроизводительные процессоры всегда имеют повышенный объем кэш-памяти.

Основные характеристики микропроцессоров Pentium приведены в таблице 8.1.

Эволюция развития МП Intel Pentium и их характеристики

Pentium. Начал выпускаться фирмой *Intel* в марте 1993 г. Имеет 32-разрядную адресную и 64-разрядную внешнюю шины данных. Микропроцессор выполнен по 0,8-микрометровой технологии, имеет более 3 млн транзисторов и работает на тактовых частотах 60 и 66 МГц. Через год появилась следующая модель микропроцессора, выполненная по 0,5-, позднее по 0,35-микрометровой технологии. Впервые был применен раздельный 16-килобайтный кэш: 8 Кбайт для инструкций и 8 Кбайт для данных. Тактовая частота была в пределах 75—200 МГц, а тактовая частота системной шины — 50—66 МГц.

Таблица 8.1

Тип процессора	Год начала выпуска	Тактовая частота первого процессора	Количество транзисторов на кристалле	Размерность внутренних регистров*	Размерность внешней шины данных	Максимальная адресуемая физическая память	Размер внутреннего кэша
8086	1978	8 МГц	29 тыс.	16GP	16	1 Мбайт	Нет
Intel 286	1982	12,5 МГц	134 тыс.	16 GP	16	16 Мбайт	Нет
Intel 386 DX	1985	20 МГц	275 тыс.	32 GP	32	4 Гбайт	Нет
Intel 486 DX	1989	25 МГц	1,2 млн	32 GP 80FPU	32	4 Гбайт	L1: 8 Кбайт
Pentium	1993	60 МГц	3,1 млн	32 GP 80FPU	64	4 Гбайт	L1: 16 Кбайт
Pentium Pro	1995	200 МГц	5,5 млн	32 GP 80FPU	64	64 Гбайт	L1: 16 Кбайт L2: 256 Кбайт или 512 Кбайт
Pentium II	1997	266 МГц	7 млн	32 GP 80FPU 64 MMX	64	64 Гбайт	L1: 32 Кбайт L2: 256 Кбайт или 512 Кбайт
Pentium III	1999	500–1000 МГц	8,2 млн	32 GP 80FPU 64 MMX 128 XMM	64	64 Гбайт	L1: 32 Кбайт L2: 512 Кбайт
Pentium IV	2000	1–4 ГГц	42 млн	32 GP 80FPU 64 MMX 128 XMM	64	64 Гбайт	L1: 16 Кбайт L2: 256, 512 Кбайт, до 8 Мбайт

* GP — регистры общего назначения, FPU — регистры математического сопроцессора

Pentium MMX – это версия *Pentium* с дополнительными мультимедиа-инструкциями (добавлено 57 новых инструкций). Основа *MMX — технология обработки множественных данных одной инструкцией (Single Instruction Multiple Data, SIMD).* Кэш-память увеличена до 32 Кбайт. Тактовая частота 166–233 МГц. Частота системной шины 66 МГц.

Pentium Pro. Промышленный выпуск начался в ноябре 1995 г. Это процессор, разработанный для 32-разрядных операционных систем. Впервые в микропроцессоре вместе с кэш-памятью L1 (здесь объемом 16 Кбайт) стали применять *кэш-память второго уровня* (L2), объединенную в одном корпусе и оперирующую на частоте микропроцессора. Выпускался сначала по 0,5-, позднее по 0,35-микрометровой технологии и имел в своем составе более 5 млн транзисторов кэш-уровня L1 объемом 16 Кбайт. Кэш уровня L2 имел объем 256, 512, 1024 и 2048 Кбайт. Тактовая частота от 150 до 200 МГц. Четырехканальная параллельная обработка данных

Частота системной шины 60—66 МГц.

Pentium Pro поддерживал все инструкции процессора *Pentium,* кроме *MMX,* а также ряд новых инструкций.

Pentium II. Первая модель микропроцессора *Pentium II* впервые появилась в мае 1997 г. Процессоры линейки *Pentium II* изготавливались по 0,35- и 0,25-микрометровой технологии. Диапазон тактовых частот от 233 до 533 МГц, дополнены ММХ-блоком.

Pentium III. Изготавливались по 0,18- и 0,13-микрометровой технологии, имеют расширенный набор *MMX (MMX2),* в основе которого лежит *технология SSE (Streaming SIMD Extentions),* где технология *SIMD* расширена на числа с плавающей запятой. Добавлены новые 128-разрядные регистры. Каждый регистр может обрабатывать четыре числа с плавающей запятой. Усовершенствована технология поточного доступа к памяти, улучшающая взаимодействие между кэш-памятью L2 и оперативной памятью. Дополнительные инструкции называются *инструкциями KNI (Katmai New Instruction).* Внедрение *KNI* предназначено для ускорения работы графических приложений и 3D-игр. Тактовая частота от 450 до 1000 МГц и выше.

Pentium IV. Отличительной особенностью этих процессоров является наличие новой системы команд-инструкций, значительно ускоряющих обработку мультимедийной информации (видео, звука, графики), — *SSE2, SSE3*. Тактовые частоты достигают 3 ГГц и выше. Двухядерные процессоры Intel Pentium Core 2 имеют частоту системной шины 1066 МГц.

8.3. Память ПЭВМ

Персональные компьютеры имеют четыре уровня памяти:

- регистровая память (микропроцессорная память процессора (МПП));
- кэш-память;
- основная память (ОЗУ и ПЗУ);
- внешняя память (ВЗУ).

Регистровая память процессора — самая быстродействующая регистровая память центрального процессора. Обеспечивает кратковременное хранение данных выполняемых команд.

Кэш-память — высокоскоростная память сравнительно большой емкости, являющаяся буфером между ОЗУ и МП и позволяющая увеличить скорость выполнения операций. Регистры кэш-памяти недоступны для пользователя, отсюда и название кэш (cache), что в переводе с английского означает «тайник».

Кэш-память (Cache Memory) или сверхоперативная память (СОЗУ) для которой используются микросхемы статической памяти. Основное назначение кэш-памяти в компьютере — служить местом временного хранения обрабатываемых в текущий момент времени кодов программ и данных.

В кэш-памяти хранятся копии блоков данных тех областей оперативной памяти, к которым выполнялись последние обращения, и весьма вероятны обращения в ближайшие такты работы — быстрый доступ к этим данным и позволяет сократить время выполнения очередных команд программы. При выполнении программы данные, считанные из оперативной памяти с небольшим опережением, записываются в кэш-память. В кэш-память записываются и результаты операций, выполненных в МП.

Кэш-память делится на уровни и, соответственно, для каждого уровня кэш-памяти используются свои, весьма различные по конструкции и быстродействию микросхемы.

Кэш первого уровня (L1) выполняется в том же кристалле, что и сам процессор, и имеет объем порядка десятков килобайт. Кэш второго уровня (L2) находится либо в кристалле процессора, либо в том же узле, что и процессор, хотя и исполняется на отдельном кристалле. Кэш-память первого и второго уровней работает на частоте, согласованной с частотой ядра процессора. Кэш-память третьего уровня выполняют на быстродействующих микросхемах и размещают на материнской плате вблизи процессора. Ее объемы могут достигать нескольких мегабайт, но работает она на частоте материнской платы.

Использование кэш-памяти существенно увеличивает производительность системы. Кэш память использует элементы статической памяти. В статической памяти элементы (ячейки) построены на различных вариантах триггеров: схем с двумя устойчивыми состояниями «0» или «1». После записи бита в такую ячейку она может пребывать в этом состоянии при наличии питания неограниченно долго.

Основная память в ПЭВМ подразделяется на ОЗУ и ПЗУ. ОЗУ (RAM – Random Access Memory) предназначено для хранения информации (программ и данных), непосредственно участвующей в вычислительном процессе в текущий интервал времени. ОЗУ — энергозависимая память: при отключении напряжения питания информация, хранящаяся в ней, теряется. Основу ОЗУ составляют микросхемы динамической памяти DRAM. Это большие интегральные схемы, содержащие матрицы полупроводниковых запоминающих элементов — полупроводниковых конденсаторов. Наличие заряда в конденсаторе обычно означает «1», отсутствие заряда — «0».

Основной составной частью микросхемы ОЗУ (рис. 8.5) является массив элементов памяти (ЭП), объединенных в матрицу накопителя. Каждый ЭП 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 элементов памяти, 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 (СМ).

По ШУ передается сигнал, определяющий, какую операцию необходимо выполнить. По ШД передается информация, записываемая в память или считываемая из нее.

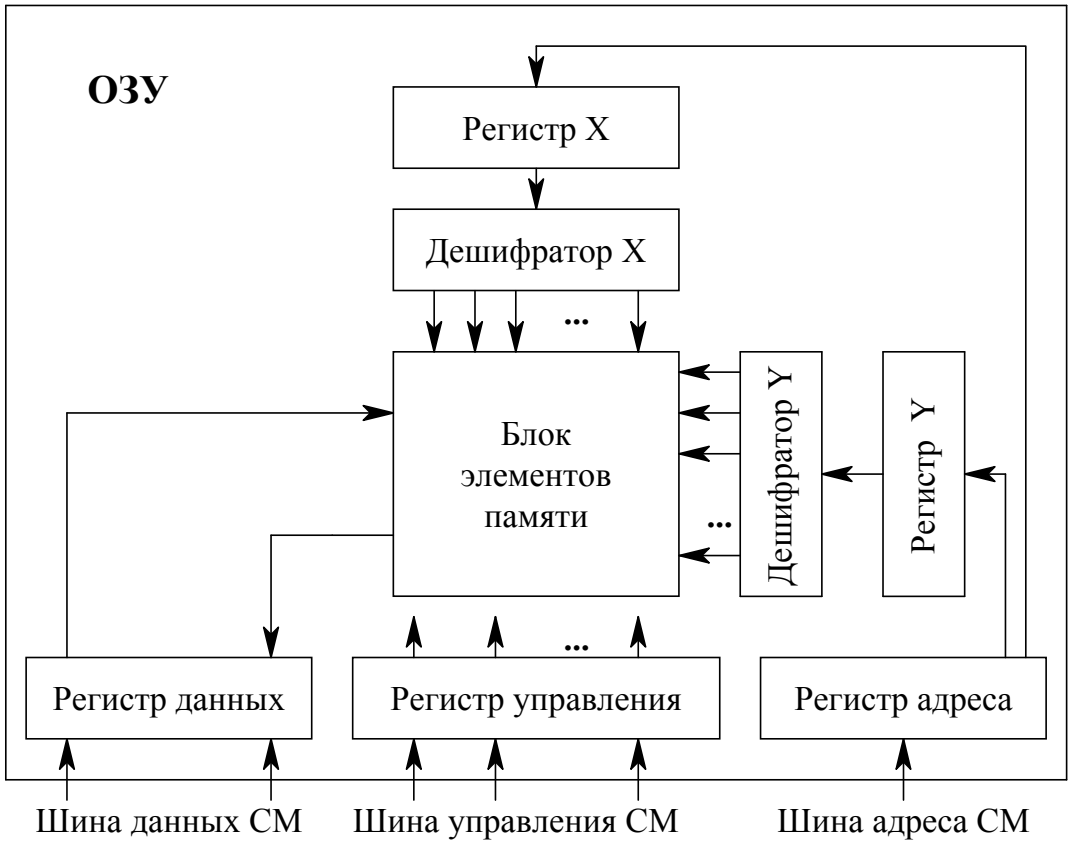


Рис. 8.5. Структурная схема ОЗУ

Отличие оперативной памяти от ВЗУ в том, что информация хранится в ней не постоянно, а временно. При выключении компьютера все содержимое оперативной памяти будет потеряно. Доступ к оперативной памяти осуществляется намного быстрее, чем к дисковой, время доступа жесткого диска составляет 8–10 миллисекунд (мс), оперативная память обладает временем доступа 3–7 наносекунд. Оперативная память используется в самых разных устройствах ПК: от видеоплаты до лазерного принтера. Микросхемы оперативной памяти в этом случае могут принадлежать к совершенно разным модификациям, однако, все они относятся к типу *динамической* оперативной памяти (DRAM).

Различают следующие типы оперативной памяти: FPM DRAM; RAM EDO; BEDO DRAM; SDRAM; DDR SDRAM; DRDRAM.

Основные характеристика типов памяти представлены в таблице 8.2.

Таблица 8.2

Тип памяти	Тактовая частота, МГц	Разрядность шины, бит	Пиковая пропускная способность, Мбайт/с
FPM DRAM	33	32	132
EDO DRAM	50	32	200
SDRAM	66	64	528
SDRAM	100	64	800
SDRAM	133	64	1064
DRDRAM, 1 канал	600	16	1200
DRDRAM, 1 канал	800	16	1600
DRDRAM, 2 канала	600	16	2400
DRDRAM, 2 канала	800	16	3200
DRDRAM, 2 канала	1066	16	4300
DDR SDRAM	100	64	1600
DDR SDRAM	150	64	2400
DDR SDRAM II	100	64	3200
DDR SDRAM II	200	64	6400

На модули памяти наносится маркировка, в которой указывается пропускная способность канала модуль-процессор, например P1600, P2100, P3200, которая рассчитывается как:

$$\text{Пропускная способность (Мбайт/с)} = \frac{\text{Fсинхронизации (МГц)} \times \text{Разрядность шины (Бит)}}{8}$$

Эта формула показывает, что производительность системы процессор-память можно увеличить, повысив тактовую частоту и увеличив разрядность шины данных.

В настоящее время оперативная память выпускается в виде микросхем, собранных в специальные модули памяти. В настоящее время применяются модули оперативной памяти емкостью 256, 512 и 1024 Мб. На большинстве материнских плат сегодня установлено три или четыре разъема для установки модулей памяти. Желательно, чтобы модули при этом обладали одной и той же скоростью доступа и были выпущены одним и тем же производителем.

ПЗУ (ROM – *Read Only Memory,* память только для чтения) предназначено для хранения загрузочных программ операционной системы, программ тестирования устройств компьютера и драйверов базовой системы ввода-вывода (BIOS).

К ПЗУ относятся энергонезависимые постоянные запоминающие устройства, из которых оперативно можно только считывать информацию, запись информации в ПЗУ выполняется при наличии специального программатора.

По технологии записи информации можно выделить ПЗУ следующих типов:

– микросхемы, программируемые только при изготовлении (классические или масочные ПЗУ или ROM);

– микросхемы, программируемые однократно – программируемые ПЗУ (ППЗУ);

– микросхемы, программируемые многократно – перепрограммируемые ПЗУ (РПЗУ).

Комплект программ, находящихся в ПЗУ, образует *базовую систему ввода-вывода (BIOS — Basic Input Output System).* Основное назначение программ этого пакета состоит в том, чтобы проверить состав и работоспособность компьютерной системы и обеспечить взаимодействие с клавиатурой, монитором, жестким диском и дисководом гибких дисков. Программы, входящие в BIOS, позволяют нам наблюдать на экране диагностические сообщения, сопровождающие запуск компьютера.

Энергонезависимая память CMOS

На материнской плате есть микросхема «энергонезависимой памяти», по технологии изготовления называемая *CMOS.* От оперативной памяти она отличается тем, что ее содержимое не стирается во время выключения компьютера, а от ПЗУ она отличается тем, что данные в нее можно заносить и изменять самостоятельно, в соответствии с тем, какое оборудование входит в состав системы. Эта микросхема постоянно подпитывается от небольшой аккумуляторной батарейки, расположенной на материнской плате.

В микросхеме *CMOS* хранятся данные о гибких и жестких дисках, о процессоре, о некоторых других устройствах материнской платы. Компьютер хранит реальное время и календарь (даже и в выключенном состоянии), это связано с тем, что показания системных часов постоянно хранятся (и изменяются) в *CMOS.*

Таким образом, программы, записанные в *BIOS,* считывают данные о составе оборудования компьютера из микросхемы *CMOS,* после чего они могут выполнить

обращение к жесткому диску, а в случае необходимости и к гибкому, и передать управление тем программам, которые там записаны.

Логическая структура основной памяти

Структурно память ПЭВМ состоит из миллионов отдельных однобайтовых ячеек памяти. Общая емкость основной памяти современных ПК обычно лежит в пределах от нескольких килобайт до 1 Гбайта и более. Емкость ОЗУ превышает емкость ПЗУ: ПЗУ занимает сотни Кбайт, остальной объем — это ОЗУ. Каждая ячейка памяти имеет свой уникальный адрес. Для ОЗУ и ПЗУ отводится единое адресное пространство.

Адресное пространство определяет максимально возможное количество непосредственно адресуемых ячеек основной памяти, логически делится на три области (рис. 8.6):

- область стандартной памяти (CMA);
- область верхней памяти (UMA);
- область расширенной памяти (HMA).

<table>
<tr><td colspan="2">Непосредственно адресуемая память</td><td colspan="2">Расширенная память (HMA)</td></tr>
<tr><td>Стандартная память (CMA)</td><td>Верхняя память (UMA)</td><td colspan="2">Высокая память</td></tr>
<tr><td>0–640 Кбайт</td><td>640–1024 Кбайт</td><td>1024–1088 Кбайт</td><td>1088–512 Мбайт</td></tr>
<tr><td>Область служебных программ и данных ОС</td><td>Служебная память область ПЗУ – BIOS</td><td colspan="2">Программы пользователей</td></tr>
</table>

Рис. 8.6. Логическая структура основной памяти

Прежде всего, основная память компьютера делится на две логические области: непосредственно адресуемую память, занимающую первые 1024 Кбайт ячеек с адресами от 0 до 1024 Кбайт, и расширенную память, доступ к ячейкам которой возможен при использовании специальных программ-драйверов или в защищенном режиме работы микропроцессора.

Драйвер — специальная программа, управляющая работой памяти или внешними устройствами компьютера и организующая обмен информацией между МП, ОП и ВУ компьютера.

Стандартная память (CMA, Conventional Memory Area, 0–640 Кбайт) предназначена для хранения:

– адресов программ обработки прерываний (векторов прерываний, 0–256 байт);

– переменных базовой системы ввода-вывода;

– ядра операционной системы.

Верхняя память (UMA, Upper Memory Area, от 640 К до 1 Мбайт, ее размер может быть равен 384 Кбайт) – эта память зарезервирована для системных целей, участки UMA имеют различное назначение, они могут быть заполнены:

– данными из буферной памяти адаптеров;

– данными из постоянной памяти;

– остаться не занятыми.

В этой области содержатся видеопамять и постоянная память BIOS, а также могут размещаться дополнительные модули постоянной и оперативной памяти, конструктивно находящиеся в адаптерах ВУ.

Расширенная память — память с адресами 1024 Кбайт и выше. В реальном режиме имеются два основных способа доступа к этой памяти:

– по спецификации *XMS* (память XMA — eXtended Memory Area);

– по спецификации *EMS* (память EMA — Expanded Memory Area).

Доступ к расширенной памяти согласно спецификации XMS *организуется при помощи специального* драйвера (например, XMM — eXtended Memory Manager) путем пересылки по мере необходимости отдельных полей XMA в свободные области верхней памяти (UMA).

Спецификация EMS *(Expanded Memory Specification)* является более ранней, согласно этой спецификации доступ реализуется не путем пересылки, а лишь путем отображения по мере необходимости отдельных полей Expanded Memory в свободные области верхней памяти. Отображение организуется путем динамического замещения адресов полей EMA в «окнах» UMA размером 64 Кбайт, разбитых

на 16-килобайтовые страницы. В окне UMA хранится не обрабатываемая информация, а лишь адреса, *обеспечивающие* доступ к этой информации. Память, организуемая по спецификации EMS, носит название *отображаемой.* Поэтому сочетание слов Expanded Memory (EM) иногда переносят как *отображаемая* память (хотя термин Expanded почти идентичен термину Extended и более точно переводится как расширенный, увеличенный). Для организации отображаемой памяти при работе в реальном режиме процессора необходим драйвер EMM.EXE (Expanded Memory Manager). Отображаемая память медленная и поэтому она постепенно уступает место Extended Memory.

В *реальном режиме расширенная память* может быть использована главным образом для хранения данных и некоторых программ ОС. Часто расширенную память используют для организации виртуальных (электронных) дисков. Исключение составляет HMA, которая может адресоваться и непосредственно при использовании драйвера HIMEM.SYS (High Memory Manager). Область HMA может использоваться для хранения любой информации, в том числе и программ пользователя.

8.4. Внешние устройства

Устройства ввода

Клавиатура — это устройство, с помощью которого осуществляется ввод данных, команд и управляющих воздействий в ПК. На клавишах нанесены буквы латинского и национального алфавитов, десятичные цифры, математические, графические и специальные служебные символы, знаки препинания, наименования некоторых команд, функций и т. д. Блок клавиатуры содержит контроллер клавиатуры, состоящий из буферной памяти и схемы управления. Он подключается к системной плате с помощью 4-проводного интерфейса (линии интерфейса используются для передачи тактовых импульсов, данных, напряжения питания +5 В, последний — «земля»). Для клавиатур существует несколько вариантов интерфейсов: стандартный разъем DIN, разъем PS/2, инфракрасный порт (IrDA), интерфейс USB.

Контроллер клавиатуры осуществляет:

– сканирование (опрос) состояния клавиш;

– буферизацию (временное запоминание) кодов клавиш на время между двумя соседними опросами клавиатуры со стороны МП;

– преобразование с помощью программируемых системных таблиц (драйвера клавиатуры) кодов нажатия клавиш (SCAN-кодов) в коды ASCII;

– тестирование (проверку работоспособности) клавиатуры при включении ПК.

При нажатии и отпускании клавиши на клавиатуре в буферную память контроллера клавиатуры поступает код нажатия или отпускания (соответственно, 0 или 1) в седьмой бит байта и номер клавиши или ее SCAN-код в остальные 7 битов. При поступлении любой информации в буферную память посылается запрос на аппаратное прерывание, инициируемое клавиатурой. При выполнении прерывания SCAN-код преобразуется в код ASCII, и оба кода (SCAN-код и ASCII-код) пересылаются в соответствующее поле ОЗУ ПЭВМ.

Контроллер клавиатуры организует и автоматическое повторение клавишной операции: если клавиша нажата более 0,5 с, то генерируются повторные коды нажатия клавиши через регулярные интервалы так, как если бы вы клавишу нажимали повторно.

Графический манипулятор мышь

Мышь (mouse) представляет собой электронно-механическое устройство, с помощью которого осуществляется дистанционное управление курсором на экране монитора. При перемещении манипулятора типа мышь по столу или другой поверхности на экране монитора соответствующим образом передвигается и курсор. Принцип работы мыши основан на преобразовании вращательного движения шарика по двум осям через оптический или электрический конвертор в серию цифровых сигналов (импульсов), пропорциональных скорости передвижения.

В настоящее время выпускаются мыши с интерфейсами COM, PS/2, USB и IrDA. Мыши с интерфейсом IrDA (инфракрасный порт) бесшнуровые и передают сигналы на приемник, подключенный к компьютеру, с помощью лучей инфракрасного диапазона.

Сканеры

Сканер — это устройство ввода в компьютер информации непосредственно с бумажного документа. Это могут быть тексты, схемы, рисунки, графики, фотографии

и другая информация. Сканер, подобно копировальному аппарату, создает копию изображения бумажного документа, но не на бумаге, а в электронном виде — формируется электронная копия изображения. Светочувствительный датчик сканера с определенной частотой производит замеры интенсивности отраженного оригиналом света. Разрешающая способность сканера прямо пропорциональна частоте замеров. В процессе сканирования устройство выполняет преобразование величины интенсивности в двоичный код, который передается в ЭВМ для дальнейшей обработки.

Разрешающая способность сканеров измеряется в количестве различаемых точек на дюйм изображения и составляет от 75 до 1600 dpi (dots per inch) и более.

Основные характеристики сканеров:

– *оптическое разрешение* — определяется как количество светочувствительных элементов в сканирующей головке, поделенное на ширину рабочей области. Выражается в точках на дюйм;

– *интерполяционное (программное, логическое) разрешение* – произвольно выбранное разрешение, для получения которого драйвер сканера рассчитывает недостающие точки;

– *разрядность (глубина цвета)* — определяет степень подробности информации об отсканированной точке изображения. Чем больше разрядов (битов) используется для представления отдельной точки изображения, тем более подробна информация о ней.

Дигитайзеры

Дигитайзер (digitaizer), или *графический планшет,* — это устройство, главным назначением которого является оцифровка изображений. Он состоит из двух частей: *основания* (планшета) и *устройства целеуказания* (пера или курсора), перемещаемого по поверхности основания. При нажатии на кнопку курсора его положение на поверхности планшета фиксируется и координаты передаются в компьютер.

Дигитайзер может быть использован для ввода рисунка, создаваемого пользователем, в компьютер: пользователь водит пером-курсором по планшету, но изображение появляется не на бумаге, а фиксируется в графическом файле.

Устройства вывода

Дисплей (монитор) — устройство визуального отображения текстовой и графической информации без ее долговременной фиксации.

Для обозначения этого типа устройств с учетом функций, выполняемых ими в ПК, в английском языке используются как синонимы термины *monitor display, monitor, video monitor, video display.* Соответственно в русском переводе также в качестве синонимов используется ряд терминов, таких как монитор, видеомонитор, видеодисплей. Мониторы бывают алфавитно-цифровые и графические, монохромные и цветного изображения.

В современных ПЭВМ в качестве устройств отображения информации наиболее широкое распространение получили цветные графические дисплеи на базе электронно-лучевых трубок, газоразрядные и жидкокристаллические дисплеи.

Дисплей на базе электронно-лучевой трубки. Основной элемент такого дисплея – электронно-лучевая трубка (рис. 8.7), а принцип его работы аналогичен принципу работы телевизора. Формирование изображения производится на внутренней поверхности экрана, покрытого слоем люминофора — вещества, светящегося под воздействием электронного луча, генерируемого специальной «электронной пушкой» и управляемого системами горизонтальной и вертикальной развертки.

Люминофор наносится в виде наборов точек трех основных цветов: красного, зеленого и синего. Эти цвета называют основными, потому что их сочетаниями (в различных пропорциях) можно представить любой цвет спектра.

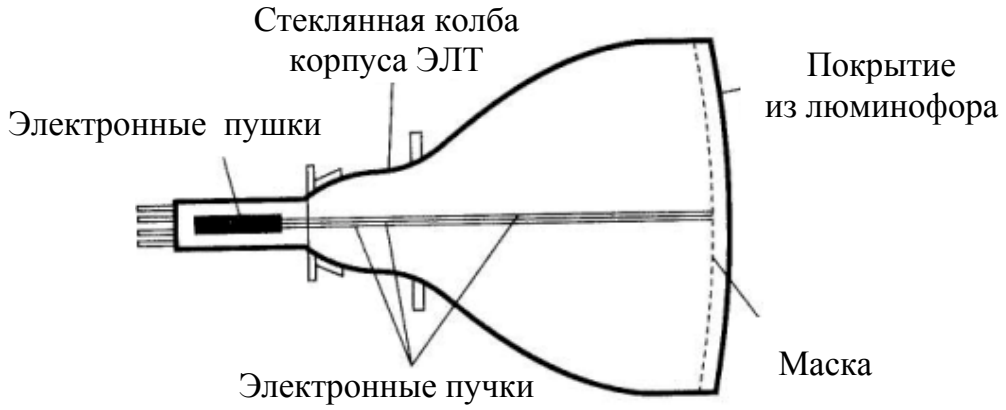


Рис. 8.7. Схема электронно-лучевой трубки

Наборы точек люминофора располагаются по треугольным триадам (рис. 8.8).

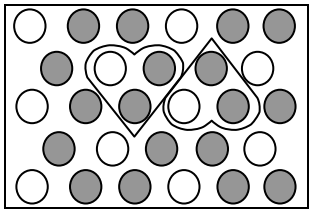


Рис. 8.8. Пиксельные триады

Триада образует пиксель (от англ. *pixel – picture element* – элемент картинки) — точки, из которых формируется изображение.

Расстояние между центрами пикселей называется точечным шагом монитора. Это расстояние существенно влияет на четкость изображения. Чем меньше шаг, тем выше четкость. В современных цветных мониторах шаг составляет 0,24 мм и менее. При таком шаге глаз человека воспринимает точки триады как одну точку «сложного» цвета.

На противоположной стороне трубки расположены три (по количеству основных цветов) электронные пушки. Все три пушки «нацелены» на один и тот же пиксель, но каждая из них излучает поток электронов в сторону «своей» точки люминофора. Чтобы электроны беспрепятственно достигали экрана, из трубки откачивается воздух, а между пушками и экраном создается высокое электрическое напряжение, ускоряющее электроны. Перед экраном на пути электронов ставится маска – тонкая металлическая пластина с большим количеством отверстий, расположенных напротив точек люминофора. Маска обеспечивает попадание электронных лучей только в точки люминофора соответствующего цвета.

Величиной электронного тока пушек и, следовательно, яркостью свечения пикселей управляет сигнал, поступающий с видеоконтроллера.

Количество отображенных строк в секунду называется строчной частотой развертки, а частота, с которой меняются кадры изображения, называется кадровой частотой развертки. Кадровая частота развертки должна быть такой, чтобы глаз человека не замечал последовательной смены кадров. Ассоциация видеоэлектронных стандартов (*VESA — Video Electronics Standards Association)* рекомендует частоту не ниже 75 кадров в секунду).

Синхросигналы строчной и кадровой разверток, а также сигналы управления яркостью лучей формируются видеоконтроллером, часто называемым видеокартой, или видеоадаптером. Основным компонентом видеокарты (рис. 8.9) является память, где хранятся передаваемые процессором числа, характеризующие каждый пиксель монитора. Цифроаналоговые преобразователи преобразуют эти числа в аналоговые сигналы, необходимые для работы монитора. Для ускорения процесса обработки видеоданных и разгрузки при этом центрального процессора ЭВМ современные видеокарты имеют свой собственный видеопроцессор.

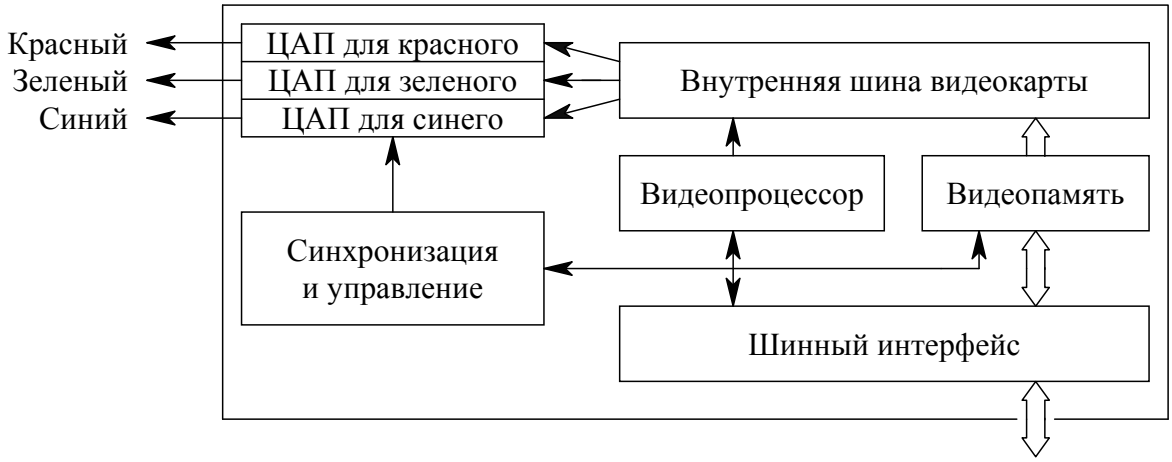


Рис. 8.9. Структура видеоконтроллера

Минимальный размер видеопамяти определяется количеством цветов и разрешающей способностью монитора. Так, для представления на мониторе 16,7 млн цветов (цвет каждого пикселя задается 24-разрядным числом) с разрешающей способностью 640×480 пикселей необходимый объем видеопамяти равен 0,9 Мбайт, при разрешающей способности 800×600 пикселей – 1,4 Мбайт.

Газоразрядные и жидкокристаллические дисплеи. Такие дисплеи часто называют панелями. *Газоразрядную панель* образуют два плоскопараллельных стекла, между которыми размещены миниатюрные газоразрядные элементы. В инертном газе газоразрядного элемента под действием управляющих сигналов, формируемых микропроцессором устройства синхронизации и подаваемых на прозрачные электроды одного или обоих стекол, возникает разряд с ультрафиолетовым излучением. Это излучение вызывает свечение нанесенного на переднее или заднее стекло люминофора одного цвета черно-белой панели или люминофоров красного, зеленого или синего цветов цветной панели.

Основой *жидкокристаллической панели* служат также две плоскопараллельные стеклянные пластины. На одну из них нанесены прозрачные горизонтальные и вертикальные токопроводящие электроды. В местах их пересечения укреплены пленочные транзисторы, два вывода которых соединены с электродами на стекле, а третий образует обкладку конденсатора. Вторую пластину конденсатора представляет прозрачный металлизированный слой на второй стеклянной пластине, расположенной параллельно первой на расстоянии, измеряемом микронами. Между пластинами помещено органическое вещество (жидкий кристалл), поворачивающее под действием электрического поля плоскость поляризации проходящего через него света. С двух сторон панели укреплены поляроидные пленки (поляризующие), плоскости поляризации которых повернуты на 90° относительно друг друга.

Растр телевизионного изображения формируется сигналами, генерируемыми устройством синхронизации и подаваемыми на электроды стеклянных пластин. При подаче на эти электроды напряжения в точке их пересечения конденсатор заряжается и возникает электрическое поле между соответствующими обкладками конденсатора. В зависимости от величины напряжения изменяется угол поляризации жидкого кристалла между обкладками конденсатора. При отсутствии напряжения и соответственно электрического поля жидкий кристалл поворачивает угол поляризации света от лампы подсветки на 90°, в результате чего свет свободно проходит через поляроидные пленки. В зависимости от напряжения на обкладках конденсатора угол поляризации может изменяться от 90° до 0°, а прозрачность ячейки панели — от максимальной до непропускания света. Панель цветного дисплея содержит красный, зеленый и синий светофильтры, образующие триаду элемента разложения изображения.

Плоские панели имеют преимущества перед вакуумными кинескопами по техническим параметрам, экологической безопасности и сроку службы. Экран современных жидкокристаллических мониторов (панелей) с разрешением 1280×1024 имеет до 5 млн точек, каждая из которых управляется собственным транзистором. Такие мониторы занимают в 2—3 раза меньше места, чем мониторы с ЭЛТ, во столько же раз легче, потребляют гораздо меньше электроэнергии и не излучают электромагнитных волн, воздействующих на здоровье людей.

Принтеры

Печатающие устройства (принтеры) — это устройства вывода данных из компьютера, преобразующие ASCII-коды и битовые последовательности в соответствующие им символы и фиксирующие их на бумаге.

Матричные принтеры

В матричных принтерах изображение формируется из точек ударным способом. В *игольчатых* (ударных) матричных принтерах печать точек осуществляется тонкими иглами, ударяющими бумагу через красящую ленту. Каждая игла управляется собственным электромагнитом. Печатающий узел перемещается в горизонтальном направлении листа, и знаки в строке печатаются последовательно. Многие принтеры выполняют печать как при прямом, так и при обратном ходе. Количество иголок в печатающей головке определяет качество печати.

Достоинства матричных принтеров: низкая стоимость расходных материалов для него, возможность одновременной печати нескольких копий.

Недостатки: невысокие качество и скорость печати, а также шум при печати.

Струйные принтеры

Струйные принтеры в печатающей головке вместо иголок имеют тонкие трубочки – сопла, через которые на бумагу выбрасываются мельчайшие капельки красителя (чернил). Матрица печатающей головки обычно содержит от 12 до 64 сопел (дюз).

Технически процесс распыления состоит в том, что в стенку сопла встроен электрический нагревательный элемент, температура которого при подаче электрического импульса резко возрастает за 5–10 мкс. Все чернила, находящиеся в контакте с нагревательным элементом, мгновенно испаряются, что вызывает резкое повышение давления, под действием которого чернила выстреливаются из сопла на бумагу. После «выстрела» чернильные пары конденсируются, в сопле образуется зона пониженного давления и в него всасывается новая порция чернил. Технологии термической пузырьковой печати придерживается большинство фирм-производителей принтеров, в том числе Canon, Hewlett-Packard, Lexmark и т. д.

Лазерные принтеры

Лазерные принтеры обеспечивают наиболее качественную печать с наивысшим разрешением и скоростью. В них применяется электрографический способ формирования изображений. Выпускаются лазерные принтеры двух основных

модификаций: *лазерные* и *светодиодные.* В лазерных принтерах для создания сверхтонкого светового луча, вычерчивающего на поверхности предварительно заряженного светочувствительного барабана контуры невидимого точечного электронного изображения, служит лазер. В светодиодных принтерах роль источника лазерного луча выполняет светодиодная панель.

С засвеченных на поверхности барабана лучом лазера или светодиодами точек стекает электрический заряд. После проявления электронного изображения порошком красителя (тонера), налипающего на разряженные участки, выполняется печать — перенос тонера с барабана на бумагу и закрепление изображения на бумаге разогревом тонера до его расплавления.

Достоинства лазерных принтеров: высокая скорость печати, высокое качество печати, бесшумность.

Недостаток лазерных принтеров – большое потребление электроэнергии и значительная стоимость расходных материалов.

8.5. Системные и периферийные шины ПЭВМ

Все каналы передачи данных, применяемые в ПК, можно условно разделить на две группы – внутренние (шины) и внешние (интерфейсы). Шины данных применяются для соединения компонентов системной платы и подключения плат расширения, а интерфейсы – для подключения внешних относительно системной платы или ПК в целом устройств накопителей, устройств ввода-вывода, коммуникационного оборудования и др.

Системные шины

Шина (bus) – совокупность линий связи, по которым информация передается одновременно. Под основной, или *системной,* шиной обычно понимается шина между процессором и подсистемой памяти. Шины характеризуются разрядностью и частотой. Важнейшими функциональными характеристиками системной шины являются количество обслуживаемых ею устройств и ее *пропускная способность,* то есть максимально возможная скорость передачи информации. Пропускная способность шины зависит от ее разрядности (есть шины 8-, 16-, 32- и 64-разрядные) и тактовой частоты, на которой шина работает.

Основные характеристики системных шин представлены в таблице 8.3.

Таблица 8.3

Характеристика	Шина ISA	EISA	MCA	VLB	PCI	AGP
Разрядность шины для данных и адреса (бит)	16 / 24	32 / 32	32 / 32	32 / 32 64 / 64	32 / 32 64 / 64	32 / 32 64 / 64
Рабочая частота (МГц)	8	8–33	10–20	до 33	до 66	66 / 133
Пропускная способность (Мбайт/с)	16	33	76	132	132 / 264 / 528	528 / 1056 / 2112
Число подключаемых устройств (HIT)	6	15	15	4	10	1

Разрядность или *ширина* шины (bus width) – количество линий связи в шине, то есть количество битов, которое может быть передано по шине одновременно.

Тактовая частота шины (bus frequency) – частота, с которой передаются последовательные биты информации по линиям связи.

Шина расширения ISA (Industry Standard Architecture – промышленная стандартная архитектура) – использовалась в первом компьютере IBM PC и позволяла добавлять в систему различные устройства.

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

Шина PC/AT –16-разрядная шина данных и 24-разрядная шина адреса, рабочая тактовая частота 8 МГц (16 МГц), имеет 7 линий для аппаратных прерываний и 4 канала DMA. Использовалась в МП 80286, совместима и с микропроцессором тактовой частотой больше 66 МГц. Благодаря 24-разрядной шине адреса адресное пространство увеличилось с 1 Мбайт до 16 Мбайт. Пропускная способность шины данных равна 5,5 Мбайт/с, ввиду ряда особенностей ее использования.

Шина EISA (Extended ISA) – 32-разрядная шина данных и 32-разрядная шина адреса создана как функциональное и конструктивное расширение ISA. Адресное пространство шины 4 Гбайт, работает на частоте 8–10 МГц.

Улучшена система прерываний, поддерживается *Bus Mastering* – режим единоличного управления шиной со стороны любого из устройств на шине. Имеет *систему арбитража* для управления доступом устройств к шине. Обеспечивается автоматическое конфигурирование системы и управление DMA. Шина поддерживает многопроцессорную архитектуру вычислительных систем и применяется в скоростных ПК, сетевых серверах и рабочих станциях.

Шина MCA (Micro Channel Architecture) – 32-разрядная шина, пропускная способность 76 Мбайт/с, рабочая частота 10–20 МГц. Шина MCA используется не очень широко из-за несовместимости плат адаптеров ISA с MCA.

Шина VLB (VL-bus, VESA Local Bus) является расширением внутренней шины МП для связи с видеоадаптером и жестким диском, платами мультимедиа, сетевым адаптером. Разрядность шины для данных – 32 бита, для адреса – 30, реальная скорость передачи данных по VLB – 80 Мбайт/с.

Шина PCI *(Peripheral Component Interconnect,* соединение внешних компонентов) – самый распространенный и универсальный интерфейс для подключения различных устройств. Шина PCI допускает подключение до 10 устройств, имеет свой адаптер, позволяющий ей настраиваться на работу с любым МП от 80486 до современных Pentium. Тактовая частота PCI – 33 МГц, разрядность – 32 разряда для данных и 32 разряда для адреса, теоретическая пропускная способность 132 Мбайт/с, а в 64-битовом варианте – 264 Мбайт/с.

Конфигурация системы с шиной PCI показана на рис. 8.10.

Конструктивно разъем шины на системной плате состоит из двух следующих подряд секций по 64 контакта. С помощью этого интерфейса к материнской плате подключаются видеокарты, звуковые карты, модемы, контроллеры SCSI и другие устройства. Как правило, на материнской плате имеется несколько разъемов PCI. Шина PCI, хотя и является локальной, выполняет и многие функции шины расширения. Шины расширения ISA, EISA, MCA (а она совместима с ними) при наличии шины PCI подключаются не непосредственно к МП, а к самой шине PCI (через интерфейс расширения). Благодаря такому решению шина является независимой от процессора (в отличие от VLB) и может работать параллельно с шиной процессора, не обращаясь к ней за запросами. Таким образом, загрузка шины процессора существенно снижается. Например, процессор работает с системной памятью или с кэш-памятью, а в это время по сети на жесткий диск записывается информация.

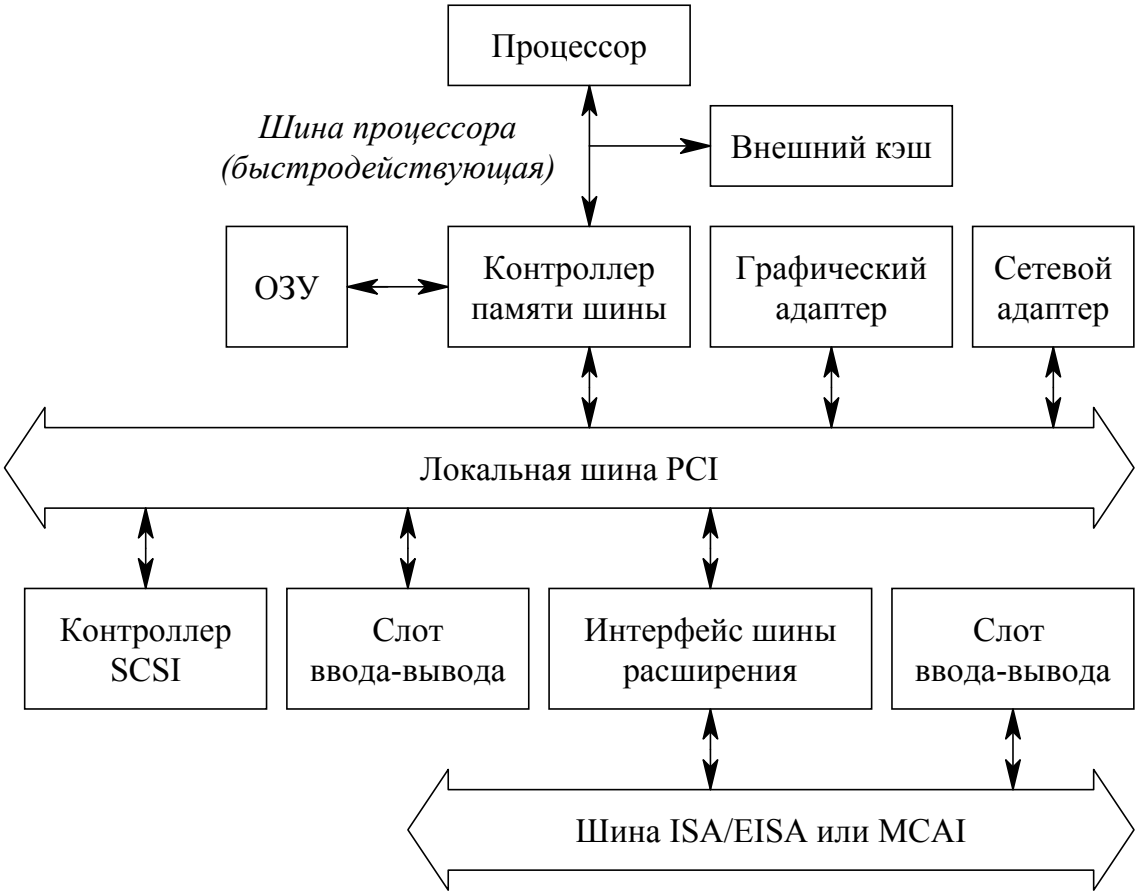


Рис. 8.10. Конфигурация системы с шиной PCI

Шина AGP *(Accelerated Graphics Port* – ускоренный графический порт) – интерфейс для подключения видеоадаптера к отдельной магистрали AGP, имеющей выход непосредственно на системную память. Шина AGP может работать с частотой системной шины до 133 МГц и обеспечивает высокую скорость передачи графических данных. Ее пиковая пропускная способность в режиме четырехкратного умножения AGP4x (передаются 4 блока данных за один такт) имеет величину 1066 Мбайт/с, а в режиме восьмикратного умножения AGP8x – 2112 Мбайт/с. По сравнению с шиной PCI, в шине AGP устранена мультиплексированность линий адреса и данных и усилена конвейеризация операций чтения-записи, что позволяет устранить влияние задержек в модулях памяти на скорость выполнения этих операций.

Конфигурация системы с шиной AGP показана на рис. 8.11.

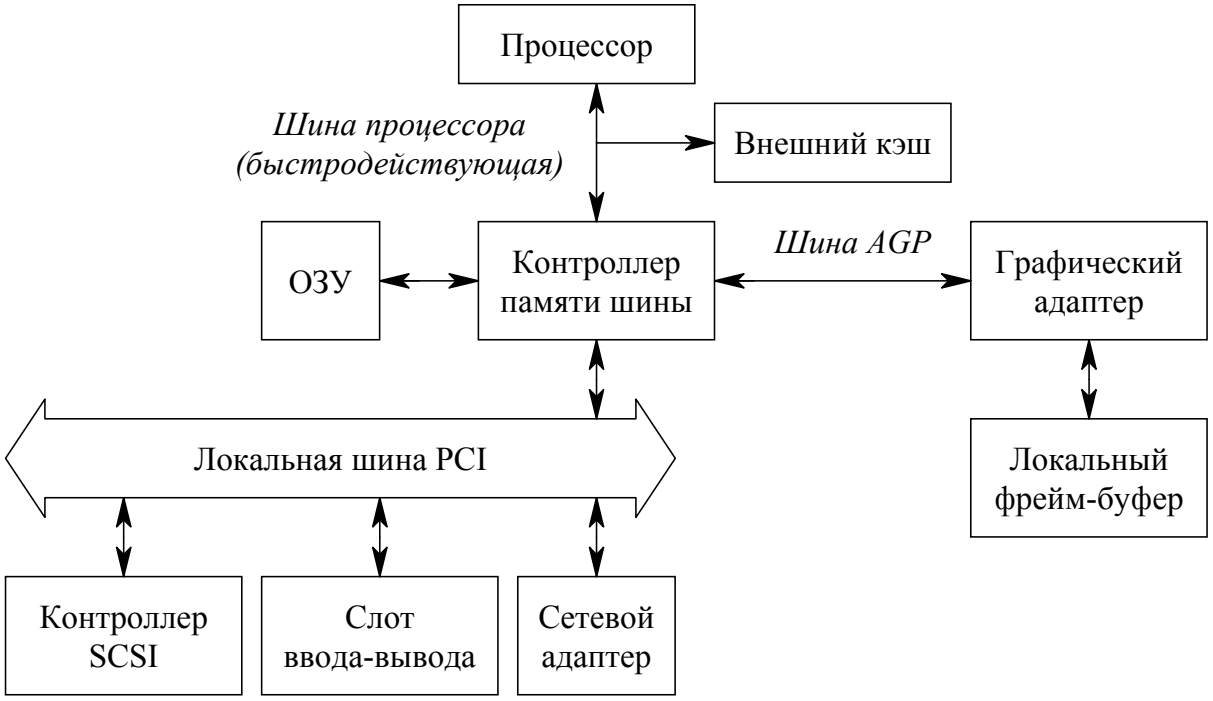


Рис. 8.11. Конфигурация системы с шиной AGP

Шина AGP имеет два режима работы: DMA и Execute. В режиме DMA основной памятью является память видеокарты. Графические объекты хранятся в системной памяти, но перед использованием копируются в локальную память карты. Обмен ведется большими последовательными пакетами. В режиме Execute системная память и локальная память видеокарты логически равноправны. Поскольку системная память выделяется динамически, блоками по 4 Кбайт, в этом режиме для обеспечения приемлемого быстродействия предусмотрен механизм, отображающий последовательные адреса фрагментов на реальные адреса 4-килобайтовых блоков в системной памяти.

PCI Express – новая архитектура соединения компонентов, введенная под эгидой PCI SIG, известная и под названием 3GIO (3-Generation Input-Output, ввод/вывод 3-го поколения). Здесь шинное соединение устройств с параллельным интерфейсом заменено на двухточечные последовательные соединения с использованием коммутаторов.

Спецификация PCI Express регламентирует последовательное соединение типа точка-точка. Масштабирование происходит увеличением не частоты, а числа каналов – PCI Express может состоять из 1, 2, 4, 8, 16 или 32 линий, данные

передаются по обоим фронтам сигнала. Пропускная способность PCI Express достигает 4 Гбайт/с в 16-канальной конфигурации. Характеристика современных шин представлена в таблице 8.4.

Таблица 8.4

	Кол-во линий	Частота, МГц	Бит за такт	Пропускная способность, Гбайт/с
AGP 3.0	32	66	8	2,1
HI 2.0	16	266	2	1,06
HT16x16	32	800	2	6,4
MuTIOL	16	133	4	1,2
PCI 32-бит	32	33	1	133
PCI 64 бит	64	66	1	533
PCI Express 32-бит	32	250	2	1,6
PCI-X	64	133	1	1,06
PCI-X 2.0	64	133	2,4	2,1–4,2

Тенденция перехода на последовательные шины еще ярче просматривается в интерфейсах для внешних устройств и накопителей. Для подключения периферийного оборудования применяются интерфейсы USB 2.0 и Fire Wire (IEEE 1394) – пропускная способность 480 и 400 Мбит/с.

Периферийные интерфейсы

Периферийные шины IDE (Integrated Drive Electronics), ATA (AT Attachment – подключаемый к AT), EIDE (Enhanced IDE), SCSI (Small Computer System Interface), SATA используются чаще всего в качестве интерфейса только для внешних запоминающих устройств.

Обычный метод обмена с жестким диском IDE – это программный ввод-вывод, при котором процессор, используя команды ввода-вывода, считывает или записывает данные в буфер жесткого диска, что отнимает какую-то часть процессорного времени. Ввод-вывод путем прямого доступа к памяти идет под управлением самого жесткого диска или его контроллера в паузах между обращениями процессора к памяти, что экономит процессорное время, но несколько снижает максимальную скорость обмена.

Существует много модификаций и расширений интерфейсов ATA/IDE. Есть интерфейсы ATA с различными номерами, Fast ATA, Ultra ATA и EIDE. Есть также IDE-интерфейсы, поддерживающие протоколы ATAPI, DMA и т. д.

Fast ATA-2 или *Enhanced IDE* (EIDE – расширенный IDE), использующий как традиционную адресацию по номерам головки, цилиндра и сектора, так и адресацию логических блоков (Logic Block Address – LBA), поддерживает емкость диска до 2500 Мбайт и скорость обмена до 16,7 Мбайт/с. К адаптеру EIDE, поддерживающему стандарт ATAPI, может подключаться до четырех накопителей, в том числе и CD-ROM, и накопитель на магнитной ленте.

ATAPI (ATA Package Interface) – стандарт, созданный с тем, чтобы напрямую подключать к интерфейсу ATA не только жесткие диски, но и дисководы CD-ROM, стримеры, сканеры и т. д. Версии интерфейса ATA-3 и Ultra ATA обслуживают диски большей емкости, имеют скорость обмена до 33 Мбайт/с, поддерживают технологию SMART (Self Monitoring Analysis and Report Technology – технологию самостоятельного следящего анализа и отчета), позволяющую устройствам сообщать о своих неисправностях, и ряд других сервисов. Современные версии интерфейса ATA/ATAPI-5, ATA/ATAPI-6 по протоколам UDMA66, 100, 133 обеспечивают пиковую пропускную способность 66, 100, 133 Мбайт/с, соответственно.

На материнских платах реализованы два канала IDE, к каждому из которых возможно подключение до двух устройств.

SCSI (Small Computer System Interface) является более сложным и мощным интерфейсом и широко используется в трех версиях: SCSI-1, SCSI-2 и SCSI-3. Это универсальные периферийные интерфейсы для любых классов внешних устройств. Фактически SCSI является упрощенным вариантом системной шины компьютера, поддерживающим до восьми устройств. Такая организация требует от устройств наличия определенных контроллеров – например, в жестких дисках SCSI все функции кодирования-декодирования, поиска сектора, коррекции ошибок и т. п. возлагаются на встроенную электронику, а внешний SCSI-контроллер выполняет функции обмена данными между устройством и компьютером – часто в автономном режиме, без участия центрального процессора.

Интерфейсы SCSI-1 имеют 8-битовую шину; SCSI-2 и SCSI-3 – 16- или 32-битовую и рассчитаны на использование в мощных машинах-серверах и рабочих станциях. Существует много различных спецификаций данного интерфейса, отличающихся пиковой пропускной способностью, максимальным числом

подключаемых устройств, предельной длиной кабеля. Пропускная способность может достигать 80, 160 Мбайт/с и более. Все SCSI-устройства управляются специальным SCSI-контроллером, реализованным чаще в виде отдельной платы расширения, устанавливаемой в свободный разъем на материнской плате. Выпускаются и материнские платы со встроенными контроллерами SCSI.

RS-232 – интерфейс обмена данными по последовательному коммуникационному порту (COM-порту). Управление работой COM-портов осуществляется специальной микросхемой UART16550A, расположенной на материнской плате. Физически разъем COM-порта может быть 25- или 9-контактным. С помощью данного интерфейса осуществляется работа и подключение таких устройств, как внешний модем, мышь и т. д.

IEEE 1284 *(Institute of Electrical and Electronic Engineers 1284* – стандарт Института инженеров по электротехнике и электронике 1284) – стандарт, описывающий спецификации параллельных скоростных интерфейсов SPP (Standard Parallel Port – стандартный параллельный порт), EPP (Enhanced Parallel Port – улучшенный параллельный порт), ECP (Extended Capabilities Port – порт с расширенными возможностями), как правило, используемых для подключения через параллельные порты компьютера (LPT-порты) таких устройств, как принтеры, внешние запоминающие устройства, сканеры, цифровые камеры. Со стороны LPT-порта установлен стандартный разъем DB-25 (25 контактов), а со стороны устройства используется разъем типа Centronics. Контроллер параллельного порта размещен на материнской плате.

Serial ATA. В конце 2000 года группа компаний Working Group (Intel, IBM, Maxtor, Quantum, Seagate и др.) анонсировала новый чрезвычайно эффективный последовательный интерфейс Serial ATA, обеспечивающий пропускную способность от 1500 Мбит/с по 8-жильному кабелю (вместо 80-жильного, используемого параллельным ATA). Интерфейс SATA широко применяется в компьютерах Pentium 4 для подключения винчестеров.

Разъемы и соединительные кабели Serial ATA благодаря малому количеству контактов весьма компактны, что упрощает подключение накопителей и благоприятно сказывается на организации охлаждения компонентов ПК. Контроллеры Serial ATA вводятся в качестве дополнительных некоторыми изготовителями системных

плат, а разработчики наборов микросхем уже предложили модификации «южных мостов», совместимые с этим интерфейсом. Тем временем организация Serial ATA Working Group, отвечающая за разработку этого стандарта, уже опубликовала спецификацию второй версии последовательного дискового интерфейса – Serial ATA-II, обеспечивающего пропускную способность до 300 Мбайт/с и полную обратную совместимость с первой версией Serial ATA (табл. 8.5).

Таблица 8.5

	Пропускная способность, Мбайт/с
ATA133	133
Ultra320 SCSI	320
FibreChannel	200
Serial ATA	150
Serial ATA-II	300
USB 2.0	60
FireWire (IEEE 1394)	50

Универсальные последовательные периферийные шины

USB *(Universal Serial Bus)* – универсальная последовательная шина, появилась в 1995 году и призвана заменить такие устаревшие интерфейсы, как RS-232 (COM-порт) и параллельный интерфейс IEEE 1284 (LPT-порт), то есть прийти на смену последовательным и параллельным портам – все устройства подключаются к одному разъему, допускающему установку многочисленных устройств и позволяет производить замену, устройств без необходимости выключения и перезагрузки компьютера. После физического подсоединения устройства правильно опознаются и автоматически конфигурируются. Шина USB самостоятельно определяет, что именно подключили к компьютеру, какой драйвер и ресурсы понадобятся устройству. Для адекватной работы шины необходима операционная система, которая корректно с ней работает. В данном случае такой ОС является Windows 98, 2000, XP. К шине USB можно одновременно подключить до 127 устройств, практически любых: мониторы, принтеры, сканеры, клавиатуры и т. д. Каждое устройство, подключенное на первом уровне, может работать в качестве коммутатора – то есть к нему, при наличии соответствующих разъемов, могут подключаться еще несколько устройств. Обмен

по интерфейсу – пакетный, скорость обмена – 12 Мбит/с. В 2001 году появилась следующая спецификация интерфейса USB 2.0 (начальный стандарт USB 1.1), обеспечивающая пропускную способность 480 Мбит/с. Поддерживается также дополнительный подканал со скоростью обмена данными в 1,5 Мбит/с для медленных устройств (клавиатуры, мыши, модема).

Шина USB реализует как синхронный (нужный, например, при проведении телеконференций), так и асинхронный режимы передачи данных.

IEEE 1394 *(Institute of Electrical and Electronic Engineers 1394* – стандарт Института инженеров по электротехнике и электронике 1394) – новый и перспективный последовательный интерфейс, предназначенный для подключения внутренних компонентов компьютера и внешних устройств. IEEE 1394 известен также под именем Fire Wire. Цифровой последовательный интерфейс FireWire характеризуется высокой надежностью и качеством передачи данных, его протокол поддерживает гарантированную передачу критичной по времени информации, обеспечивая прохождение видео- и аудиосигналов в реальном масштабе времени без заметных искажений. При помощи шины FireWire можно подсоединить друг к другу огромное количество различных устройств по технологии Plug and Play и практически в любой конфигурации, чем она выгодно отличается от названных ранее трудно конфигурируемых шин типа SCSI. К одному контроллеру возможно подключение до 63 устройств на один порт с помощью единого шестижильного кабеля.

Пропускная способность интерфейса составляет от 100–400 Мбит/с до 1600 Мбит/с. Этот интерфейс будет использоваться для подключения жестких дисков, дисководов CD-ROM и DVD-ROM, а также высокоскоростных внешних устройств, таких как цифровые видеокамеры, видеомагнитофоны и т. д.

PCMCIA *(Personal Computer Memory Card International Association* – ассоциация производителей плат памяти для персональных компьютеров) – внешняя шина компьютеров класса ноутбуков. Другое название модуля PCMCIA – PC Card. Шина имеет разрядность 16/26 (данные/адрес, адресное пространство – 64 Мбайт), поддерживает автоконфигурирование, возможно подключение и отключение устройств в процессе работы компьютера. Конструктов – миниатюрный 68-контактный разъем. Контакты питания сделаны более длинными, что позволяет вставлять и вынимать карту при включенном питании компьютера.

ACPI *(Advanced Configuration Power Interface* – расширенный интерфейс конфигурирования и питания) – интерфейс, представляющий собой единую систему управления питанием для всех компонентов компьютера. Поддерживается новейшими модификациями BIOS материнских плат.

Радиоинтерфейс Bluetooth

В 1998 г. разработана технология беспроводной передачи данных, которая получила название Bluetooth. Новый интерфейс позволил соединять друг с другом практически любые устройства: ноутбуки, принтеры, цифровые фотоаппараты, мобильные телефоны. Для беспроводного интерфейса Bluetooth отведен частотный диапазон от 2,4 до 2,48 ГГц. Поскольку радиоэфир имеет помехи естественного и искусственного происхождения, то для использования в интерфейсе Bluetooth был предложен принцип скачкообразной перестройки частоты в пределах отведенного диапазона по псевдослучайному алгоритму.

Дальность надежного соединения для устройств с интерфейсом Bluetooth составляет от 10 до 100 м и более. Скорость передачи данных в асимметричном режиме до 721 Кбит/с, а в симметричном – 432,6 Кбит/с в обоих направлениях. Для целей безопасности в спецификации интерфейса предложено использовать аутентификацию и шифрование данных с ключом длиной от 8 до 128 битов.

Контрольные вопросы

1. Состав системного блока ПЭВМ.
2. Что такое чипсет (ChipSets)?
3. Что входит в архитектуру системной платы?
4. Какие функции выполняет микропроцессор?
5. Основные характеристики микропроцессора.
6. Виды памяти в ПЭВМ?
7. Назовите назначение ОЗУ и ПЗУ.
8. Логическая структура основной памяти ПЭВМ.
9. Принцип работы клавиатуры.
10. Основные типы и характеристики системных шин ПЭВМ.

9. ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

9.1. Классификация и характеристики внешних запоминающих устройств

Внешние запоминающие устройства (ВЗУ) обеспечивают долговременное хранение программ и данных. ВЗУ классифицируются по ряду признаков (рис. 9.1): виду носителя, типу конструкции, принципу записи и считывания информации, методу доступа. При этом под *носителем* понимается материальный объект, способный хранить информацию.

В зависимости от типа носителя все ВЗУ можно подразделить на магнитные накопители, оптические накопители и полупроводниковую память (флэш-память).

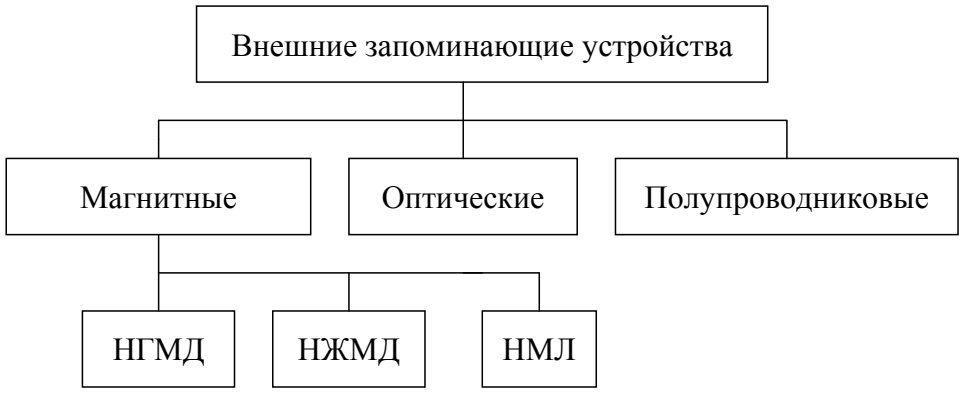


Рис. 9.1. Классификация ВЗУ

Кроме основной своей характеристики – *информационной емкости* – дисковые накопители характеризуются и двумя другими показателями (табл. 9.1):

- временем доступа;
- скоростью считывания последовательно расположенных байтов.

Время доступа (access time) к информации на диске, то есть время, которое дисковод затрачивает до начала чтения-записи данных, складывается из нескольких составляющих:

- времени перемещения магнитной головки на нужную дорожку (seek time);
- времени установки головки и затухания ее колебаний (setting time);
- временем ожидания вращения (rotation latency) – ожидания момента, когда из-за вращения диска нужный сектор окажется под головкой.

Таблица 9.1

Тип накопителя	Емкость, Мбайт	Время доступа, мс	Скорость считывания, Кбайт/с
FDD	1,2; 1,44	65 – 100	55 – 150
HDD	1 000 – 250000	5 – 30	500 – 6000
DVD	4700 – 17000	150 – 200	1380
CD-ROM	250 – 1500	50 – 300	150 – 3000
CD-RW	120 – 1000	50 – 150	150 – 3000
НМОД	128 – 2600	50 – 150	300 – 6000

Диски относятся к машинным носителям информации с прямым доступом. Понятие *прямой доступ* применительно к диску означает, что ПК может «обратиться» к дорожке, на которой начинается участок с искомой информацией или куда нужно записать новую информацию, непосредственно, где бы ни находилась головка чтения-записи накопителя. После доступа к информации происходит ее последовательное считывание (*скорость считывания (transfer rate) 2 Мбайт/с и выше).*

Флэш-память относится к классу EEPROM (Electrical Erasable Programmable Read Only Memory) – электрически перезаписываемая постоянная память. Флэш-память обладает сочетанием высокой плотности информации (ячейки меньше ячеек DRAM), энергонезависимого хранения, электрического стирания и записи и высокой надежности.

Флэш-память имеет время доступа 35–200 нс. Стирание информации занимает 1–2 секунды, запись байта занимает время порядка 10 мкс.

9.2. Накопители на гибких магнитных дисках

В общем случае под накопителем на магнитных дисках понимают устройство, обеспечивающее запись и считывание данных с вращающихся магнитных дисков.

Магнитный диск – носитель информации в форме круглой пластины (диска), поверхность которой покрыта магнитным материалом.

Принцип записи цифровой информации на магнитный диск заключается в следующем (рис. 9.2). Дисковод вращает диск под магнитной головкой, которая может

двигаться по радиусу диска, при этом каждое ее положение создает на диске кольцевой путь – дорожку. Количество дорожек определяется числом различных положений головки. Информация записывается на диске вдоль дорожки путем подачи на головку тока записи, который создает магнитный поток, проходящий через зазор головки и магнитный слой диска. Изменением направления сигнала в обмотке головки изменяют полярность намагничивания. Считывание информации происходит за счет индуцирования тока в обмотке головки при перемещении под головкой намагниченных участков дорожки.

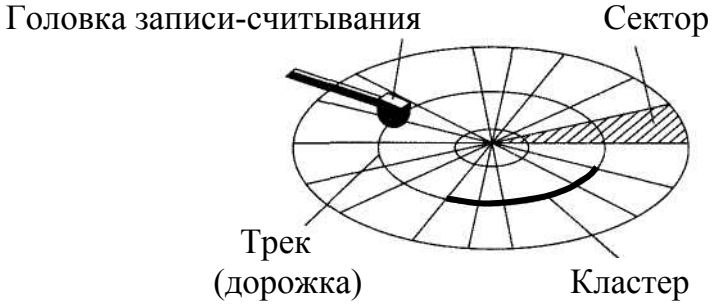


Рис. 9.2. Магнитная запись цифровой информации

Гибкий магнитный диск, состоящий из круглой полимерной подложки, покрытой с обеих сторон тонким слоем магнитного материала и помещенной в специальную пластиковую упаковку, называется *дискетой.* Дискеты используются для хранения данных и переноса их между компьютерами, оснащенными НГМД.

Конструктивно НГМД состоит из четырех основных элементов:

– *рабочий двигатель,* обеспечивающий постоянную скорость вращения дискеты (300 об./мин);

– *рабочие головки,* предназначенные для записи и чтения данных. Дисковод оснащается двумя комбинированными головками (для чтения и записи каждая), которые располагаются над рабочими поверхностями дискеты – одна головка предназначена для верхней, а другая – для нижней поверхности дискеты;

– *шаговые двигатели,* предназначенные для движения и позиционирования головок;

– *управляющая электроника,* отвечающая за передачу и преобразование информации, которую считывают или записывают головки.

Дискета устанавливается в дисковод, автоматически в нем фиксируется, после чего механизм накопителя раскручивается до номинальной частоты вращения. В накопителе вращается дискета, магнитные головки остаются неподвижными. При этом дискета вращается только при обращении к ней (чтении или записи данных). Чтобы не нарушалась постоянная скорость вращения привода, дисковод всегда должен работать только в горизонтальном или вертикальном положении. Процессор взаимодействует с НГМД через специальный контроллер гибких дисков.

Необходимое условие использования дискеты для записи и чтения информации – ее форматирование, т. е. разбиение (разметка) на определенные участки, по номерам которых можно определить любую запись на диске.

Для форматирования дисков операционные системы используют специальные команды: для DOS – это команда Format. Дискета разбивается на дорожки (треки), а дорожки на сектора (рис. 9.2). *Сектор* представляет собой минимальную физическую единицу хранения информации на диске, его размер, как правило, равен 512 байт. Секторы на дорожке записываются последовательно. Между каждым сектором остается промежуток, предназначенный для синхронизации. Дорожки нумеруются начиная от края к центру диска, при этом каждая дорожка имеет одно и то же количество секторов. Таким образом, на дорожках, расположенных ближе к центру диска, информация записывается более плотно. Дорожки с одинаковыми номерами на различных поверхностях диска (в общем случае – пакета дисков) образуют *цилиндр.* Доступ к информации записанной в одном цилиндре, осуществляется без перемещений магнитных головок.

В настоящее время наибольшее распространение получили 3,5-дюймовые (89 мм) дискеты высокой плотности DS/HD (*double-side /high-density* – две стороны, высокая плотность). Для них число дорожек на одной стороне равно 80, количество секторов на дорожках – 18, соответственно, емкость диска $80 \cdot 18 \cdot 2 \cdot 512 = 1\,474\,560$ байт, или 1 440 Кбайт, или 1,44 Мбайт.

Основные характеристики магнитного диска приведены в табл. 9.2.

Основные функции управления дисководом реализуются встроенным в материнскую плату контроллером FDD. Он осуществляет кодирование информации, поиск дорожек и секторов, синхронизацию, коррекцию ошибок.

Таблица 9.2

Параметр	ГМД 3,5 дюйма (89 мм)
Полная емкость (Кбайт)	1600
Рабочая емкость (Кбайт) после форматирования	1440
Плотность записи (бит/мм)	558
Плотность дорожек (дорожек/мм)	5,3
Число дорожек на одной стороне диска	80
Число поверхностей (сторон)	2
Среднее время доступа (мс)	65
Скорость передачи (Кбайт/с)	150
Скорость вращения (оборотов/мин)	360
Число секторов	18
Емкость сектора дорожки (байт)	512

Структура записи данных вдоль дорожки диска показана на рис. 9.3.

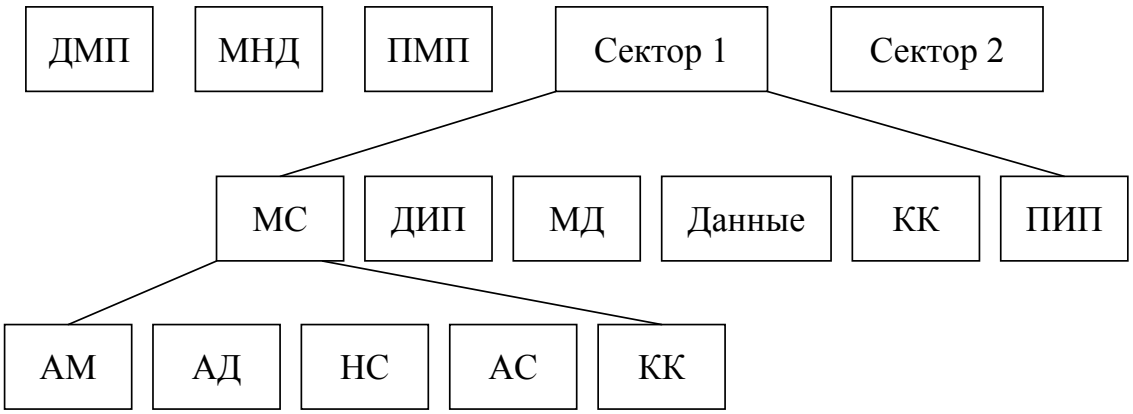


Рис. 9.3. Структура данных на дорожке гибкого диска

Данные содержат служебную и рабочую (данные пользователя) информацию. Служебная информация необходима для синхронизации работы НГМД. Она идентифицирует:

– дорожку и включает домаркерный промежуток (ДМП), маркер начала дорожки (МНД), послемаркерный промежуток (ПМП);

– сектор и включает маркер сектора (МС), доинформационный промежуток (ДИП), маркер данных (МД), данные, контрольные коды (КК), послеинформационный промежуток (ПИП);

– маркер сектора содержит адресный маркер (АМ), адрес дорожки (АД), номер стороны дискеты (НС), адрес сектора на дорожке (АС) и контрольные коды (КК).

Все промежутки заполняются нулевыми байтами и используются для создания временных интервалов при чтении-записи, необходимых системе управления. Маркер или метка – это определенный для каждого признака идентифицирующий код. Контрольные коды вводятся для проверки достоверности информации на дорожке.

9.3. Накопители на жестких магнитных дисках

В накопителе на жестких магнитных дисках носители информации представляют собой круглые жесткие пластины (называемые также платтерами), обе поверхности которых покрыты слоем магнитного материала. Первая подобная система памяти была создана фирмой IBM в 1956 г. и называлась RAMAC 305 (Random Access Method of Accounting and Control). Данное запоминающее устройство состояло из 50 алюминиевых дисков (покрытых магнитным слоем) диаметром около 60 см и толщиной 2,5 см, которые были насажены на ось мощного электромотора. На поверхности каждого диска располагалось 100 концентрических дорожек, на каждой из которых можно было запомнить 500 алфавитно-цифровых символов, закодированных в использовавшемся тогда семибитном коде. RAMAC 305 состоял из двух огромных блоков, занимающих площадь 3–3,5 м, и мог хранить 5 млн символов.

Современные НЖМД строятся по винчестерской технологии и называются винчестерами. Данная технология впервые была применена при создании накопителей на жестких дисках (модели IBM 3340) на предприятии IBM в английском городе Винчестер в 1973 г. В винчестерах головки считывания-записи вместе с их несущей конструкцией и дисками заключены в герметичный закрытый корпус. Головка, используемая в винчестере, имеет небольшие размеры и массу и размещается на держателе специальной аэродинамической формы. При вращении диска над ним образуется тонкий воздушный слой, обеспечивающий «воздушную подушку» для зависания головки над поверхностью диска на расстоянии доли микрон. При этом масса головки

и прижимающее усилие к поверхности диска настолько малы, что, даже если в процессе работы устройства головка опускается на зону данных, вероятность их повреждения очень низкая.

Винчестер (рис. 9.4) состоит из нескольких одинаковых дисков, расположенных друг под другом. Для каждого диска в винчестере имеется пара рабочих головок, которые приводятся в движение и позиционируются шаговым двигателем. Все головки расположены «гребнем». Позиционирование одной головки обязательно вызывает аналогичное перемещение и всех остальных, поэтому, когда речь идет о разбиении винчестера, обычно говорят о цилиндрах (cylinder), а не о дорожках. Цилиндр – это совокупность всех совпадающих друг с другом дорожек по вертикали, по всем рабочим поверхностям.

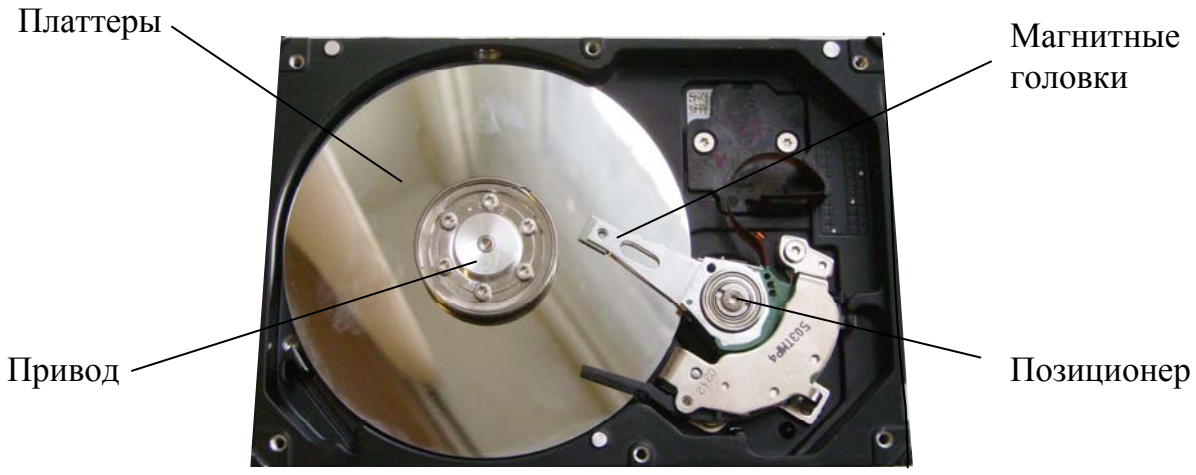


Рис. 9.4. Накопитель на жестких магнитных дисках

Общая емкость пакета дисков НЖМД определяется произведением количества цилиндров (C), количества магнитных головок (H), количества секторов на дорожке (S) и размера сектора в байтах (как правило, 512 байт).

$$\text{Объем винчестера} = C \cdot H \cdot S \cdot 512 \text{ байт}.$$

Например, винчестер емкостью 1,2 Гбайт содержит 2632 цилиндра с 16 магнитными дорожками на каждом цилиндре и с 63 секторами на дорожке.

Современные винчестеры имеют очень высокую плотность хранения информации – порядка миллиардов бит на квадратный сантиметр – и очень большую емкость:

от десятков до сотен Гбайт. Кроме объема, основными характеристиками производительности накопителя являются:

– *время доступа* – интервал между моментом, когда процессор запрашивает с диска данные, и моментом их выдачи. Время доступа зависит от расположения головок и пластин под ними, поэтому для него даются средние значения, составляющие в настоящее время единицы миллисекунд;

– *частота вращения* – частота, с которой пластины диска вращаются относительно магнитных головок (измеряется в об./мин). У современных моделей скорость вращения шпинделя (вращающего вала) составляет порядка 10 000 об./мин;

– *средняя скорость передачи данных диском,* определяемая временем для передачи данных после запуска операции чтения. Эта скорость зависит также от канала ввода-вывода. У современных моделей максимальная скорость передачи данных составляет порядка 300 Мбайт/с;

– *размер кэш-памяти.* Все современные накопители снабжаются встроенным *кэшем* (порядка 10 Мбайт), который существенно повышает их производительность.

На быстродействие винчестера оказывает сильное влияние то, как размещены секторы на дорожках и соседних сторонах дисков. Если все секторы будут идти друг за другом и параллельно на каждой стороне диска, то скорость доступа к информации будет не слишком велика, т. к. электроника, которая считывает данные с диска, имеет ограниченное быстродействие. Для увеличения быстродействия еще в самых первых винчестерах использовался метод *чередования секторов* (Interleave), при котором номера секторов идут не последовательно. В настоящее время в винчестерах используется буферная память, что позволяет считать сразу всю дорожку, поэтому чередование секторов на дорожке теперь не всегда применяют, а для секторов на соседних дорожках такой способ остался в силе, при этом чаще используется сдвиг между началом дорожек на разных сторонах.

9.4. Накопители на оптических дисках

К оптическим накопителям относятся устройства:

CD-ROM (Compact Disk Read Only Memory) – неперезаписываемые лазерно-оптические диски или компакт-диски ПЗУ;

CD-R (Compact Disk Recordable) – компакт-диски с однократной записью;

CD-RW (CD Rewritable) – компакт-диски перезаписываемые, с многократной записью;

DVD-ROM (Digital Versatile Disk Read Only Memory) – неперезаписываемые цифровые универсальные диски;

DVD-R (DVD Recordable) – цифровые универсальные диски с однократной записью;

DVD-RW (DVD Rewritable или DVD-RAM – DVD Read Access Memory) – цифровые перезаписываемые универсальные диски;

CC-WORM (Continuous Composite Write Once Read Many) – магнитооптические накопители с однократной записью;

CC-E (Continuous Composite Erasable) – перезаписываемые магнитооптические накопители.

В устройствах CD-ROM носителем информации является *оптический диск* (компакт-диск), изготавливаемый на поточном производстве с помощью штамповочных машин и предназначенный только для чтения.

Компакт-диск 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. Толщина напыления составляет несколько десятитысячных долей миллиметра.

Информация на диске представляется в виде последовательности впадин (углублений в диске или питов) и выступов (их уровень соответствует поверхности диска), расположенных на спиральной дорожке, выходящей из области вблизи оси диска. На каждом дюйме (2,54 см) по радиусу диска размещается 16 тыс. витков спиральной дорожки. Для сравнения – на поверхности жесткого диска на дюйме по радиусу помещается лишь несколько сотен дорожек. Емкость такого CD достигает 800 Мбайт.

В отличие от магнитных дисков, компакт-диски имеют не множество кольцевых дорожек, а одну – спиральную.

Первое поколение CD-ROM имели скорость передачи данных равную 150 Кбит/с. В следующих поколениях CD-ROM скорость вращения постепенно увеличивали, а для классификации устройств использовали кратность относительно первого поколения. В табл. 9.3 приведено соотношение между кратностью привода и скоростью передачи данных CD-ROM и DVD.

Таблица 9.3

Кратность	CD	DVD
1^x	150 Кбайт/с	1,32 Мбайт/с
4^x	600 Кбайт/с	5,28 Мбайт/с
6^x	900 Кбайт/с	7,93 Мбайт/с
8^x	1,2 Мбайт/с	10,57 Мбайт/с
10^x	1,5 Мбайт/с	13,21 Мбайт/с
12^x	1,8 Мбайт/с	15,84 Мбайт/с
16^x	2,4 Мбайт/с	21,13 Мбайт/с
20^x	3,0 Мбайт/с	26,4 Мбайт/с
32^x	4,8 Мбайт/с	42,26 Мбайт/с

При записи лазерная оптическая головка устанавливается механизмом позиционирования на дорожку, соответствующую адресу (рис. 9.5).

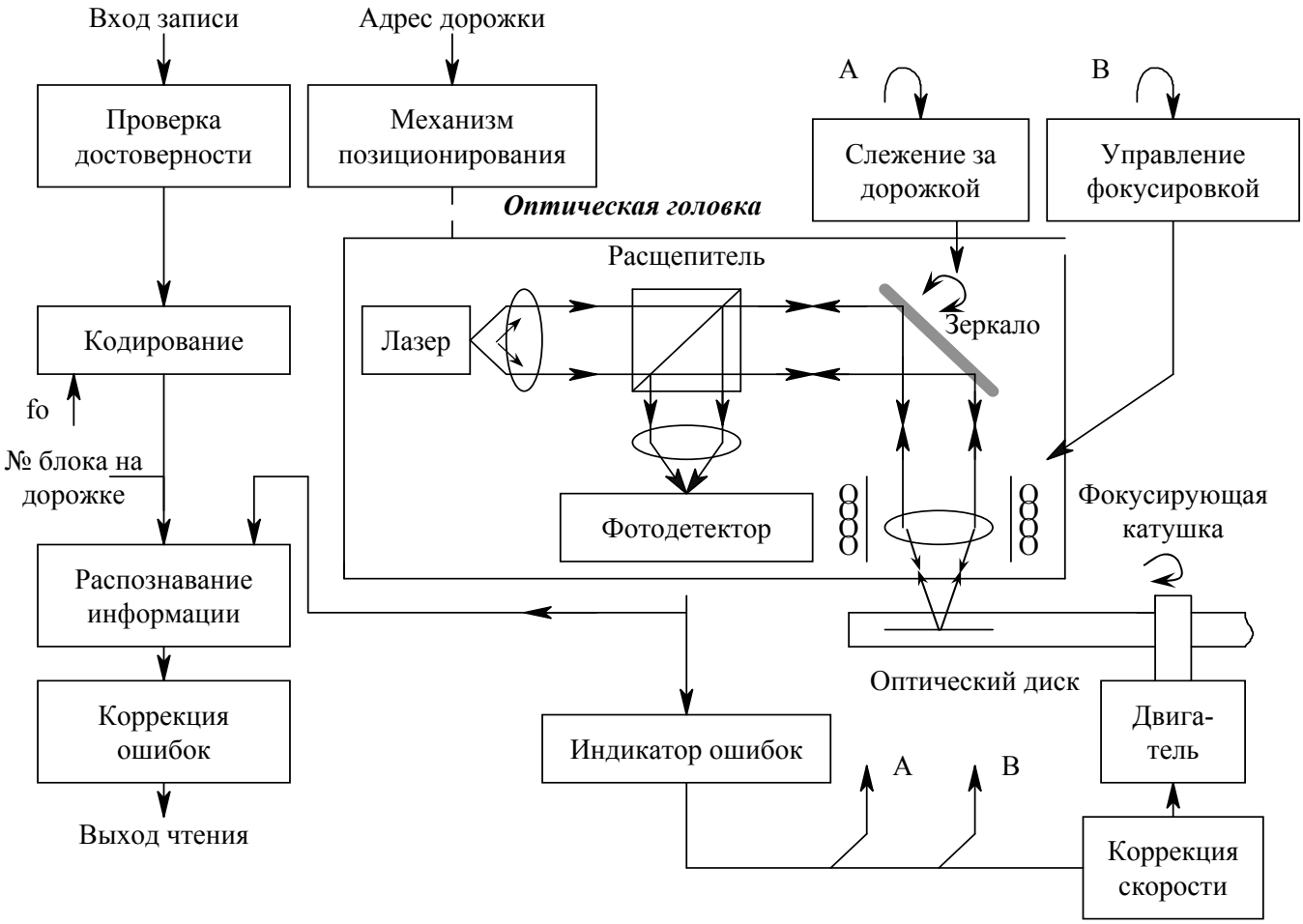


Рис. 9.5. Схема оптико-механического лазерного блока

Со служебной дорожки считываются сигналы на индикатор ошибок. Он вырабатывает сигналы, корректирующие скорость двигателя дисковода, фокусировку луча лазера и его положение на дорожке.

Информация, подлежащая записи, поступает на вход записи. Она проверяется на достоверность, перекодируется и дополняется разрядами, обеспечивающими обнаружение и исправление ошибок при чтении. Сформированный таким образом информационный сигнал поступает на вход модуляции записывающего луча лазера. Модулированный луч лазера последовательно проходит конденсаторную линзу, расщепитель (коллиматор), отражается от зеркала слежения за дорожкой, проходит фокусирующие линзу и катушку и попадает на запоминающую пленку оптического диска, прожигая в ней отверстие (при записи 1) или оставляя неповрежденной (при записи 0).

В режиме чтения требуемого (по адресу) блока информации на вход диодного лазера поступает небольшое постоянное напряжение, так что лазер генерирует луч невысокой постоянной интенсивности, который не может прожечь отверстие на пленке диска. Отражаясь от элементов дорожки, луч приобретает модуляцию по интенсивности и поступает обратно по тому же пути до расщепителя, из которого выходит через линзу на фотодетектор. Далее он поступает на блок распознавания информации с помощью выделенных синхроимпульсов и на блок коррекции ошибок. В последнем блоке с помощью дополнительных разрядов, введенных при записи, обнаруживается и исправляется большинство ошибок.

Фокусировка луча лазера на отражательной поверхности диска осуществляется с помощью объектива.

Таким образом, если в последовательных тактах считывания длина пути света не меняется, то и состояние светодиода остается прежним. В результате ток через светодиод образует последовательность двоичных электрических сигналов, соответствующих сочетанию впадин (0) и выступов (1) на дорожке (рис. 9.6).

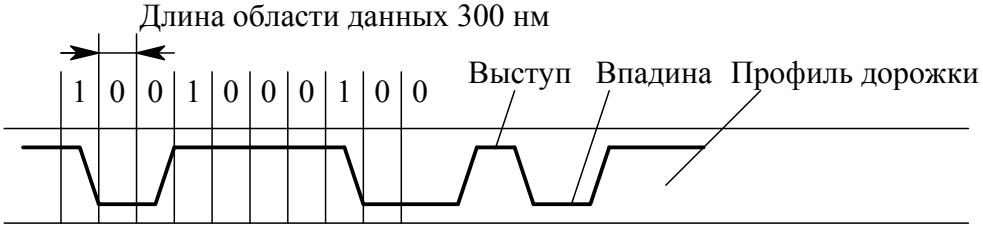


Рис. 9.6. Профиль дорожки CD-ROM

Различная длина оптического пути луча света в двух последовательных тактах считывания информации соответствует двоичным единицам. Одинаковая длина соответствует двоичным нолям.

Накопители CD-R позволяют считывать информацию с компакт-дисков и записывать на оптические диски CD-R с помощью лазерного луча.

Запись на такие диски осуществляется благодаря наличию на них особого светочувствительного слоя из органического материала, темнеющего при нагревании. В процессе записи лазерный луч нагревает выбранные точки слоя, которые темнеют и перестают пропускать свет к отражающему слою, образуя участки, аналогичные впадинам.

Накопители CD-RW дают возможность делать многократную запись на диск, которая производится гораздо более сложным способом. В этом случае применяется специальный комбинированный слой красителя, который при нагреве лазерным лучом способен многократно менять свои характеристики. Вещество такого слоя при этом может многократно переходить из кристаллического состояния в аморфное и обратно.

Накопители DVD имеют тот же геометрический размер, что и обычные CD-ROM, но вмещают до 17 Гбайт данных. Это достигнуто благодаря уменьшению размеров углублений (питов) и ровных участков, а также расстояния между дорожками, использованию лазеров с меньшей длиной волны (635 нм против 780 нм у дисководов CD-ROM). Кроме того, DVD-диски могут иметь два слоя информации (оба слоя имеют отражающую поверхность, при этом один из них обладает высокой прозрачностью – до 40 %), к которым можно обращаться, регулируя фокусировку лазерного луча.

Магнитооптический накопитель использует для записи и стирания информации специальный магнитный слой, который реагирует как на оптическое, так и на магнитное воздействие. Запись информации осуществляется с помощью лазера, нагревающего отдельные участки слоя до температуры выше 150 °C, при которой может изменяться ориентация намагниченности. После этого магнитной головкой на диск записываются данные. При считывании информации вектор поляризации лазерного луча, отраженного от слоя, меняет свое направление в зависимости от ориентации намагниченного участка.

Преимуществом магнитооптического метода записи по сравнению с магнитным является независимость от внешних магнитных полей при нормальных температурах, поскольку перемагничивание возможно только при температуре выше 150 °С.

9.5. Структура файловой системы Windows

За организацию хранения и доступа к информации на магнитном носителе, как и на любом другом носителе информации, отвечает файловая система, являющаяся важной составной частью любой операционной системы. Понятие «файловая система» включает совокупность всех файлов на диске, наборы служебных структур данных, используемых для управления файлами (каталоги файлов, дескрипторы файлов, таблицы распределения свободного и занятого пространства на диске и т. п.), а также комплекс системных программных средств, предназначенных для реализации операций с файлами: поиска, чтения, записи, создания, уничтожения, установки атрибутов и уровней доступа и т. п.

Для того чтобы файловая система могла использовать жесткий диск или дискету, их необходимо отформатировать. Форматирование жесткого диска включает три этапа: низкоуровневое форматирование диска; создание главных (основных) разделов или логических дисков на расширенном разделе; логическое форматирование главных разделов или логических дисков. Процедура форматирования дискет включает два совмещенных этапа: низкоуровневое и логическое форматирование – и осуществляется за один шаг.

Низкоуровневое форматирование диска выполняется, как правило, на заводе-изготовителе. При этом определяются размер сектора, количество дорожек и количество секторов на дорожку, на диск также записывается информация о коррекции ошибок и идентификации секторов (для каждого сектора).

Жесткий диск IBM-совместимых компьютеров может содержать, как правило, до четырех *основных разделов,* каждый из которых может быть использован конкретной файловой системой. Кроме того, на диске может быть создан один так называемый *расширенный* (extended) *раздел,* который в свою очередь может разбиваться на несколько логических дисков, также используемых конкретной файловой системой. Таким образом, на диске может быть один, два или три главных раздела и один

расширенный раздел, содержащий один или несколько логических дисков. При этом в качестве системного раздела (раздела, содержащего зависимые от аппаратной платформы файлы, необходимые для загрузки и инициализации операционной системы) можно использовать только главный раздел. Главные разделы, а также каждый из логических дисков обозначаются одной из букв английского алфавита и двоеточием. Буквой *C:* обозначается первый главный раздел. Следующий получает букву *D:*, потом *E:* и т. д. (Буквой *A:* общепринято обозначать дисковод для гибких дисков, буква *B:* зарезервирована на тот случай, если в компьютере не один, а два дисковода гибких дисков.) При создании первого раздела на диске (основного или расширенного) в первом физическом секторе жесткого диска создается главная загрузочная запись (master boot record – MBR) и таблица разделов (partition table), содержащая информацию о каждом из имеющихся на диске разделов. Главная загрузочная запись используется программой начальной загрузки BIOS (ROM Bootstrap routine), которая при загрузке с жесткого диска считывает и загружает в память первый физический сектор на активном разделе диска, называемый загрузочным сектором (Boot Sector).

В процессе *логического форматирования главных разделов* или *логических дисков* на диск записывается информация, необходимая для работы конкретной файловой системы, в том числе и загрузочный сектор раздела (Partition Boot Sector).

Современные операционные системы могут работать одновременно с несколькими файловыми системами.

На рис. 9.7 представлена схема раздела файловой системы FAT. (Свое название FAT получила от одноименной таблицы размещения файлов – File Allocation Table.)

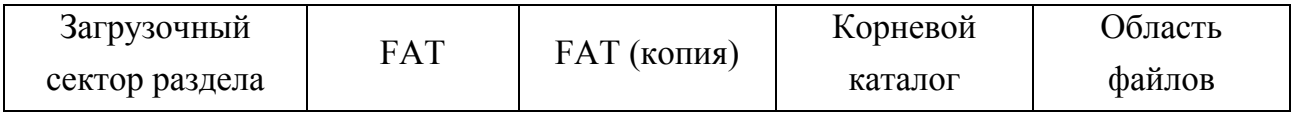


Рис. 9.7. Структура раздела FAT

Корневой каталог содержит список имен файлов с указанием даты, времени их создания и размеров. В качестве дополнительной информации каталог включает атрибуты файла: только для чтения, системный, скрытый или архивный. Кроме того, в каталоге содержится начальная позиция файла, т. е. номер первого кластера на диске, содержащего данные требуемого файла.

Таблица размещения файлов (FAT) – это список, содержащий информацию о расположении данных файла на диске. Для каждого кластера отводится один элемент списка, содержащий, помимо информации о расположении данных файла, информацию о состоянии кластера: занят, свободен, испорчен.

Когда системе нужен какой-то файл, она находит его стартовый кластер по имени файла в каталоге их размещения и затем просматривает FAT в поисках элемента списка, соответствующего начальному кластеру. Если весь файл помещен в одном кластере, то элемент FAT содержит индикатор конца файла. Если файл занимает несколько кластеров, элемент FAT указывает номер следующего кластера, в котором должно находиться продолжение файла, либо признак его окончания. В сущности, FAT содержит цепочки ссылок, следуя по которым можно найти размещение каждого файла на диске. Для предотвращения возможной потери информации таблица размещения файлов дублируется на случай повреждения первой FAT.

Размер таблицы FAT при фиксированном объеме диска зависит от размера кластера: чем меньше размер кластера, тем больше их количество и, следовательно, больше размер таблицы FAT. Таким образом, использование кластеров, размер которых больше одного сектора, помимо снижения фрагментации, уменьшает объем дискового пространства, необходимого для хранения FAT.

Первоначально для записи в таблице размещения файлов адреса любого файла FAT использовала 12 бит и поддерживала разделы объемом до 16 Мбайт. Для поддержки дисков размером более 32 Мб разрядность FAT была повышена до 16 бит – FAT 16. С помощью 16 битов можно выразить 2^{16} (65 536) разных значений. Это значит, что файлам на жестком диске не может быть предоставлено более чем 65 536 кластеров.

Таким образом, чем больше жесткий диск, тем больше места на нем расходуется впустую из-за несовершенной системы адресации файлов. Один из способов борьбы с нерациональными потерями – разбиение жесткого диска на несколько разделов, или логических дисков, каждый из которых имеет собственную таблицу размещения файлов. В итоге потери, обусловленные большими размерами кластеров, становятся меньше.

Начиная с файловой системы Windows 95 OSR2 при записи адреса файла на жестком диске используется не два, а четыре байта, или 32 бита (FAT 32). С помощью

32 битов можно выразить 2^{32} (4 294 967 296) разных значений, т. е. файлам на жестком диске может быть предоставлено 2^{32} кластеров. В этом случае размеры отдельных кластеров могут быть значительно меньше и нерациональные потери дисковой памяти уменьшаются (табл. 9.4).

Таблица 9.4

Объем диска	Количество секторов на кластер	Размер кластера
513 Мбайт… 8192 Мбайт	8	4 Кбайт
8193 Мбайт… 16 384 Мбайт	16	8 Кбайт
16 385 Мбайт… 32 768 Мбайт	32	16 Кбайт
Более 32 768 Мбайт	64	32 Кбайт

Файловая система NTFS (New Technology File System) использует кластеры в качестве фундаментальной единицы дискового пространства. При этом для записи адреса файла может использоваться 8 байт (64 бита), и соответственно, файлам на жестком диске может быть предоставлено 2^{64} кластеров. Однако на практике используются таблицы разделов размерами до 2^{32} секторов, т. е., работая с файловой системой NTFS, можно создать файл, максимальный размер которого составляет 2^{32} кластеров (как и при использовании FAT 32).

Структура раздела файловой системы NTFS представлена на рис. 9.8.

Загрузочный сектор раздела	Главная таблица файлов (MFT)	Системные файлы (метафайлы)	Область файлов

Рис. 9.8. Структура раздела NTFS

Форматирование раздела для использования файловой системы NTFS приводит к созданию нескольких системных файлов и главной таблицы файлов – файла MFT (Master File Table), содержащего информацию обо всех файлах и папках, имеющихся в разделе NTFS. Первые 16 записей MFT зарезервированы для служебных файлов, называемых также метафайлами, причем первая запись таблицы описывает непосредственно саму главную файловую таблицу – сам MFT, также являющийся метафайлом.

За ней следует запись зеркальной копии MFT, гарантирующая доступ к зеркальному файлу MFT в случае, если первая запись MFT будет разрушена. Местоположение сегментов данных MFT и зеркального файла MFT хранится в загрузочном секторе раздела, который также дублируется. С третьей по шестнадцатую записи MFT содержат описания других метафайлов, каждый из которых отвечает за какой-либо аспект работы системы. Семнадцатая и последующие записи главной файловой таблицы используются собственно файлами и каталогами на томе.

Отличительная особенность файловой системы NTFS – значительное расширение возможностей по управлению доступом к отдельным файлам и каталогам, большое число атрибутов файлов (в том числе атрибутов защищенности), позволяющих обеспечить защиту данных от несанкционированного доступа. При использовании FAT возможность установки прав доступа к отдельным каталогам и файлам отсутствует. Единственной мерой защиты служат права доступа к разделяемым ресурсам, которые устанавливаются на весь разделяемый ресурс, действуют по отношению ко всем имеющимся на нем файлам и папкам и имеют силу только при доступе через сеть.

Контрольные вопросы

1. По каким признакам классифицируются ВЗУ?
2. На какие типы классифицируются магнитные накопители?
3. Назовите характеристики дисковых накопителей.
4. Что понимается под сектором на магнитном диске и какова его емкость?
5. Какие параметры учитываются в определении емкости магнитного диска 1,44 Мбайта?
6. Какие параметры учитываются в определении емкости жесткого диска?
7. Назовите типы оптических дисков.
8. Охарактеризуйте схему раздела FAT.
9. В чем особенность файловой системы NTFS?

10. ДИАГНОСТИКА ПЭВМ

10.1. Программа POST

При включении питания ПЭВМ, аппаратному сбросу от кнопки RESET или нажатии комбинации клавиш CTRL+ALT+DEL процессор переходит к исполнению программы *POST* (Power-On Self Test), хранящейся в микросхеме ПЗУ BIOS. Программа POST выполняет тестирование процессора, памяти и системных средств ввода-вывода, а также конфигурирование всех программно-управляемых аппаратных средств системной платы.

Обязательные и наиболее важные этапы – это тестирование регистров процессора и оперативной памяти (особенно первых 64 Кбайт, где размещаются служебные регистры). После этого тестируются остальные ресурсы.

Последовательность шагов программы POST:

1. Проверка регистров процессора.
2. Проверка контрольной суммы BIOS.
3. Проверка таймера.
4. Проверка контроллеров DMA.
5. Проверка регенерации памяти и тестирование первых 64 Кбайт.
6. Проверка интерфейса клавиатуры.
7. Инициализация контроллера прерываний и установка векторов.
8. Проверка батареи и контрольной суммы CMOS.
9. Проверка защищенного режима.
10. Получение конфигурации из CMOS.
11. Проверка видеоадаптера.
12. Проверка контроллера прерываний.
13. Проверка клавиатуры.
14. Тест памяти от 64 до 640 Кбайт.
15. Тест памяти свыше 1 Мбайт.

При обнаружении какой-либо ошибки при тестировании аппаратуры BIOS информирует пользователя о неисправности звуковым сигналом или выводом

текстового сообщения. При неактивизированном видеоадаптере пользователь информируется об ошибке набором звуковых сигналов. Звуковые сигналы фатальных ошибок, при которых тестирование прекращается, а процессор переходит в режим останова, приведены в таблице 10.1. Если обнаруженная ошибка не является фатальной, например, села батарейка питания CMOS, после выдачи звукового сигнала процесс тестирования продолжается.

Таблица 10.1

Сигнал	Описание
	Фатальные ошибки аппаратуры
1 короткий	Ошибка регенерации ОЗУ
2 коротких	Ошибка четности ОЗУ
3 коротких	Ошибка в области первых 64 Кбайт ОЗУ
4 коротких	Ошибка системного таймера
5 коротких	Ошибка процессора
6 коротких	Ошибка на линии A20, которой управляет контроллер клавиатуры
7 коротких	Ошибка переключения режимов процессора
8 коротких	Ошибка при работе с видеопамятью
9 коротких	Ошибочная контрольная сумма BIOS
10 коротких	Ошибка при работе с CMOS-памятью
11 коротких	Ошибка кэша
	Нефатальные ошибки аппаратуры
1 длинный, 3 коротких	Ошибка ОЗУ
1 длинный, 8 коротких	Ошибка видеоподсистемы

В случае, если к моменту обнаружения ошибки видеоадаптер работает, то на экран монитора выводятся код ошибки и краткое описание на английском языке. Обычно это ошибки, связанные с проблемами, которые не позволят загрузить операционную систему или с неисправностью блоков на системной плате.

Когда процесс тестирования оборудования доходит до момента вывода на экран монитора сообщений, с помощью которых пользователь может следить за процессом работы программы POST, внизу экрана монитора появляется

информационная строка, в которой содержится информация о том, как перейти в режим BIOS Setup.

После тестирования и конфигурирования (настройка устройств) программа POST инициализирует загрузку операционной системы.

10.2. Главное меню программы BIOS

Главное меню программы BIOS Setup предназначено для настройки и изменения конфигурации компьютера и имеет интерфейс в виде меню. Для входа в Setup во время выполнения программы POST необходимо нажать клавишу <Del> на клавиатуре. Меню утилиты Setup, способы перемещения по пунктам и выбор параметров зависят от производителя и версии BIOS.

Набор опций расширенного BIOS Setup включает:

– первоначальную загрузку (C, A, CD-ROM), управление параметрами клавиатуры, конфигурирование кэш-памяти, периферийных устройств;

– управление параметрами процессора (частоты синхронизации), оперативной памяти, кэш-памяти;

– встроенные утилиты определения дисковых интерфейсов (IDE, SATA);

– утилиты тестирования и форматирования дисков;

– группа опций безопасности (Security) – установка пароля на вход, ограничение доступа к дискам;

– средства антивирусной защиты – предупреждение о попытке записи в Boot-сектор и проверка при загрузке на совпадение контрольной суммы с хранящимся в CMOS эталонным значением;

– группа опций управления параметрами системы энергосбережения (Power Management).

Стандартное главное меню компьютера Pentium 4 включает:

1. Standard CMOS Features – меню стандартных настроек аппаратных средств и установки системных даты и времени;

2. Advanced BIOS Features – установка стартовой конфигурации компьютера и параметров BIOS. Выбор порядка загрузки компьютера (Boot Device);

3. Advanced Chipset Features – меню настройки параметров чипсета;

4. Integrated Peripherals – меню изменения параметров периферийных устройств;

5. Power Management Setup – настройка параметров энергосбережения;

6. PnP/PCI Configurations – меню конфигурирования распределения ресурсов компьютера;

7. Load Fail-Safe Defaults и Load Optimized Defaults – меню восстановления настройки BIOS;

8. BIOS Setting Password – меню установки пароля на BIOS;

9. Save & Exit Setup – выход из BIOS Setup с сохранением новых установок;

10. Exit Without Saving – выход из BIOS Setup без сохранения (изменение конфигурации компьютера).

10.3. Тестовые программные средства

Кроме встроенных средств тестирования в ПЭВМ используются и автономные средства контроля и диагностики, которые позволяют детализировать системную информацию: определение конфигурации и характеристик ПЭВМ – тип процессора, тип материнской платы, типы используемых дисков, объем оперативной памяти и ее распределение, видеосистемы и др. Помимо контроля работоспособности ПЭВМ, программы информируют пользователя об эффективности использования ресурсов системы и осуществляют их перераспределение.

Тестовые программы используются для идентификации конфигурации компьютера, его системных ресурсов, а также для его диагностики и оценки его относительной производительности. Их можно разделить на две группы:

1. *Специализированные* – ориентированные на анализ какой-либо одной подсистемы компьютера. Например, широко используются программы для оценки производительности процессоров, жестких дисков, а также программы, тестирующие работу оперативной памяти, видеоконтроллеров, звуковых карт;

2. *Универсальные* программы, позволяющие выполнять комплексный анализ всех подсистем компьютера (Sandra, Checkit, Astra).

Тестовая программа Checkit

Checkit – это информационно-диагностическая программа, позволяющая определить системную конфигурацию ПЭВМ, протестировать основные его компоненты и оценить производительность отдельных подсистем.

В программе имеются тесты всех типов памяти (стандартной, расширенной и дополнительной), компонентов системной платы (процессора, сопроцессора, контроллеров), часов реального времени, последовательных и параллельных портов, принтеров. Для жесткого диска применяются также операции, проверяющие каждую физическую дорожку. При контроле видеоподсистемы отдельно тестируются текстовые и графические режимы, а также видеопамять. Программа предусматривает интерактивный контроль устройств ввода: клавиатуры, мыши. Могут быть выполнены: идентификация модемов, сетевых плат, приводов CD-ROM, тестирование приводов гибких дисков, оценка производительности для системной платы, жесткого диска (время доступа и скорость передачи), видеоподсистемы (скорость обмена). Интерфейс программы представлен на рис. 10.1.

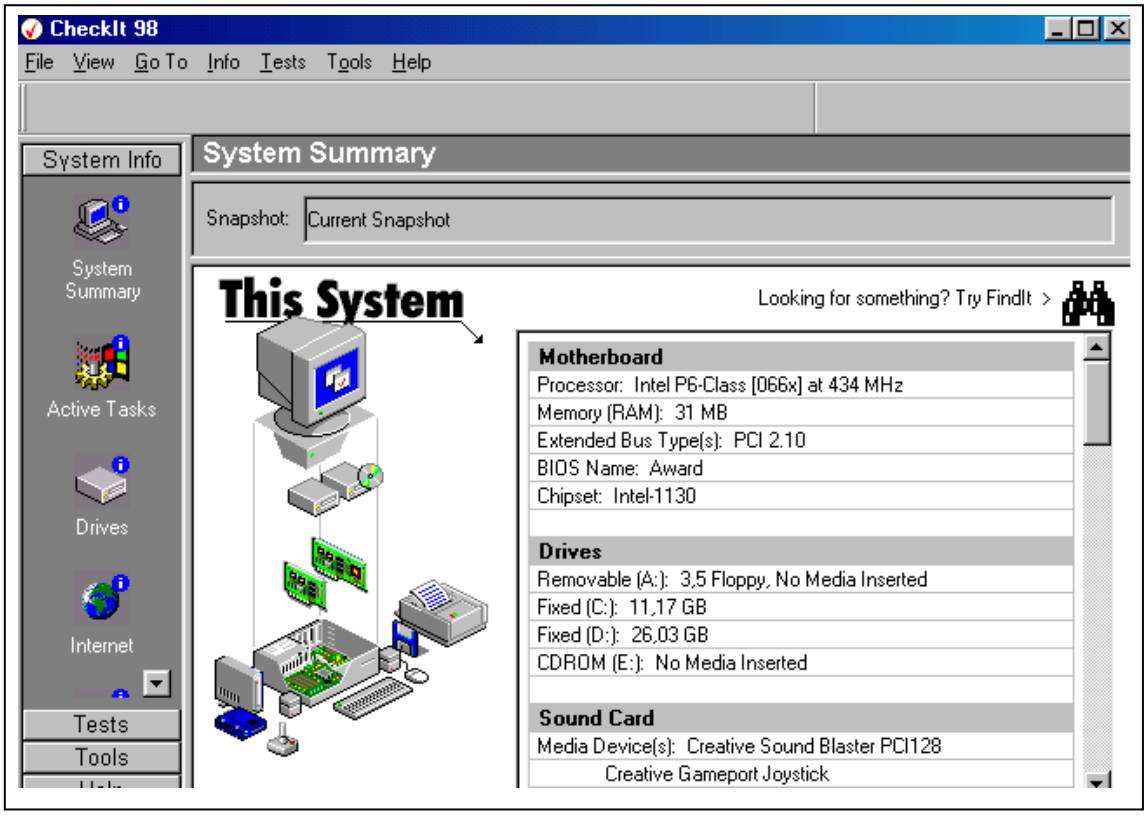


Рис. 10.1. Интерфейс программы Checkit

Программа Checkit интегрирована в систему Windows, имеет удобный интерфейс, тестирует узлы ПЭВМ и создает подробный отчет о результатах выполненных проверок. Окно программы устроено по принципу слева – оглавление, справа – содержание выбранного пункта. Программа имеет три большие группы операций, которые представлены в левом окне-оглавлении: System Info (сведения о системе), Tests (тесты), Tools (дополнительные средства).

Группа System Info

В группе System Info находятся утилита сбора информации о ПЭВМ, где содержится сведения о составе системы, драйверах, программах, настройках локальной сети и Интернет (рис. 10.1). В нем имеются следующие подменю:

System Summary – сбор информации о системе в целом;

Active Tasks – сведения о программах и драйверах (версия и предназначение).

Drives – сведения о дисках (наименование, размер, свободное место, тип файловой системы).

Motherboard – сведения о материнской плате (тип процессора, BIOS).

Memory – сведения о памяти – размер, свободное место в оперативной памяти, текущий размер виртуальной памяти (swap-файла), использование ресурсов памяти системой Windows.

Operating system – сведения о файлах операционной системы.

Ports Test – сведения о контроллерах системы.

Ports – сведения о портах (тип, назначение портов).

Video – сведения о видеосистеме (монитор, видеоадаптер, поддерживаемые режимы работы).

Группа проверок Tests

Motherboard Test – проверка материнской платы, включает проверку процессора, сопроцессора, часов реального времени, контроллеров доступа к памяти и контроллеров прерываний (рис. 10.2).

Memory Test – проверка основной и дополнительной, а также видеопамяти. Программа предложит перейти в режим перезагрузки и на черном экране протестирует оперативную память. Кроме обычной проверки памяти, имеется еще углубленная проверка – переключатель Deep Testing в секции Type of Test («вид тестирования»).

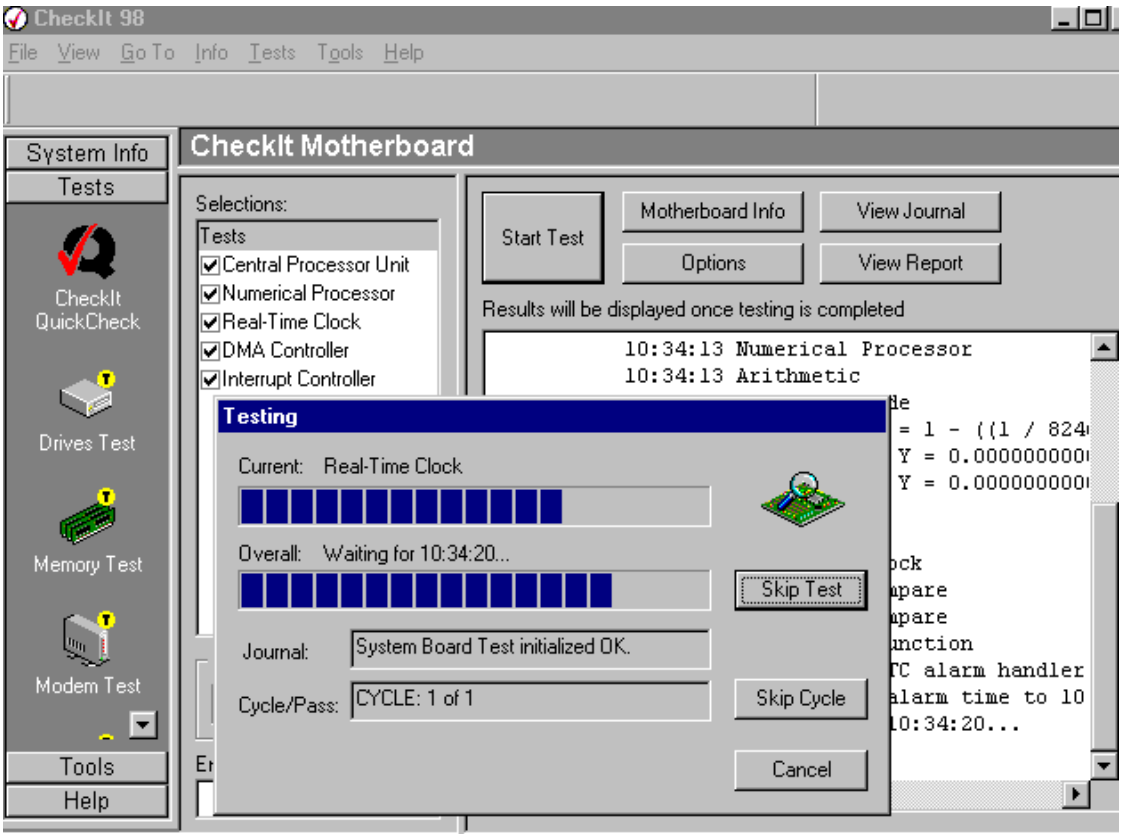


Рис. 10.2. Тестирование материнской платы

Drives Test – тест дисковой подсистемы, проверяются загрузочная область, таблицы размещения файлов и корневая директория. Тест проверяет накопители на магнитных дисках, CD-ROM и другие сменные диски. Например для проверки читаемости компакт-диска в CD-ROMe, необходимо указать проверку области данных (галочка в квадратике Read Data Sectors). Тогда программа последовательно считает с CD-диска информацию и представит отчет о результатах проверки.

Modem Test – проверка модема. Тестируются основные и дополнительные команды модема, а также специальные (для факс-модемов), скорость обмена с COM-портом.

Ports Test – проверка параллельного и последовательного портов.

Video Calibration – калибровка монитора, позволяет проверить качество монитора во всех возможных для него и видеокарты режимах.

Quick Check – проверка дерева системы. Сверху располагаются данные об оборудовании, ниже – информация о Windows (дата сохранения системных файлов, директория размещения временных файлов).

Группа утилит Tools

Утилита *Active Update* – обновление программы через Интернет (доступна при подключении).

Утилита *Benchmark* – эталонный тест, проводит проверку быстродействия ПЭВМ. В нем можно посмотреть производительность каждой из подсистем компьютера. Программа хранит результаты предыдущих проверок, что позволяет осуществлять контроль производительности системы и программного обеспечения.

Утилита *Reports* – позволяет напечатать отчет, включив в него информацию о системе (группа System Info Reports), результаты тестирования (группа Tests) и журналы тестирования (Test Journals). Кнопка Print Now отправляет отчет на принтер.

Утилита *System Spy* – осуществляет контроль за состоянием и изменениями системы. Производится сравнения производительности системы последнего отчета сохранения (Snapshot) с результатами текущего тестирования системы и сообщается, какие программные и аппаратные изменения произошли. Все что осталось неизменным, будет написано черным, все удаленное будет написано красным, все вновь установленное – зеленым, а все изменившееся – синим.

Утилита *Save Critical Files* сохраняет важные файлы Windows – те, при повреждении которых нормальная работа продолжаться не может. Создаются копии десяти файлов: системного реестра (user.dat, system.dat), командного процессора (command.com), файлов настройки DOS и Windows-режимов (autoexec.bat, config.sys, win.ini, system.ini) и некоторых других.

Утилита *Restore Critical Files*, позволяющая восстановить десять файлов, если они оказались повреждены.

Утилита *System Tune-Up* – вызывает последовательно проверку диска.

Тестовая программа Sandra

Sandra – это информационно-диагностическая программа, позволяющая получить информацию об аппаратном и программном обеспечении, произвести комплексный анализ работы устройств, входящих в состав ПЭВМ. Интерфейс программы Sandra представлен на рис. 10.3.

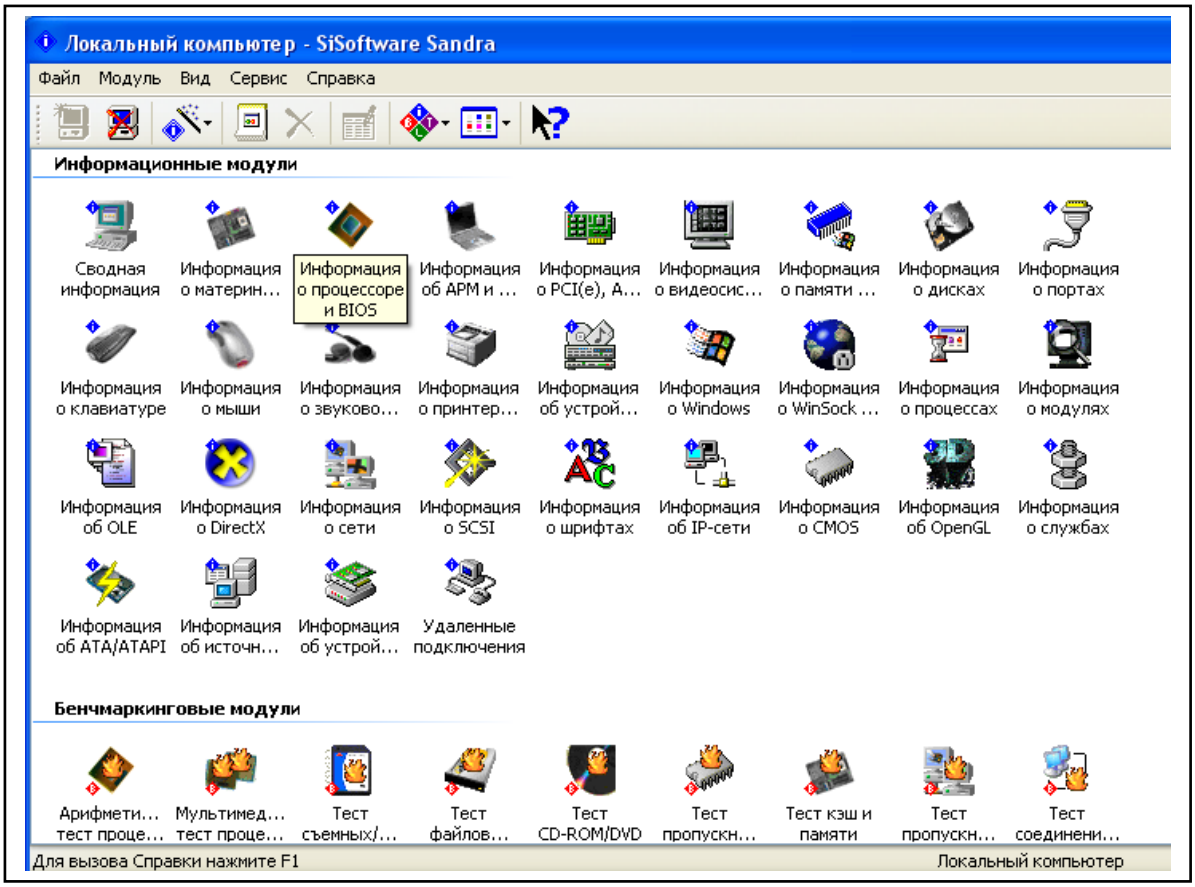


Рис. 10.3. Интерфейс программы Sandra

Возможности программы Sandra:

– получение информации о системе в целом, материнской плате, процессоре, BIOS, дисковой системы, портах, памяти, видеосистеме;

– тестирование процессора, памяти, файловой системы видеосистемы, пропускной способности сети и др.;

– сравнение производительности процессора, подсистемы памяти, файловой системы со стандартными системами и эталонными образцами;

– стресс тестирование, позволяющее определить стабильность работы системы.

Состав:

– информационные модули;

– бенчмаркинговые модули;

– тестовые модули;

– просмотровые модули;

– мастера.

Информационные модули программы Sandra (рис. 10.4) включают:

- информацию о системе;
- информацию о материнской плате;
- информацию о процессоре, КЭШе, BIOS;
- информацию о дисках;
- информацию о портах;
- информацию о памяти;
- информацию о видеосистеме;
- информацию о клавиатуре, манипуляторе, принтере и т.д.

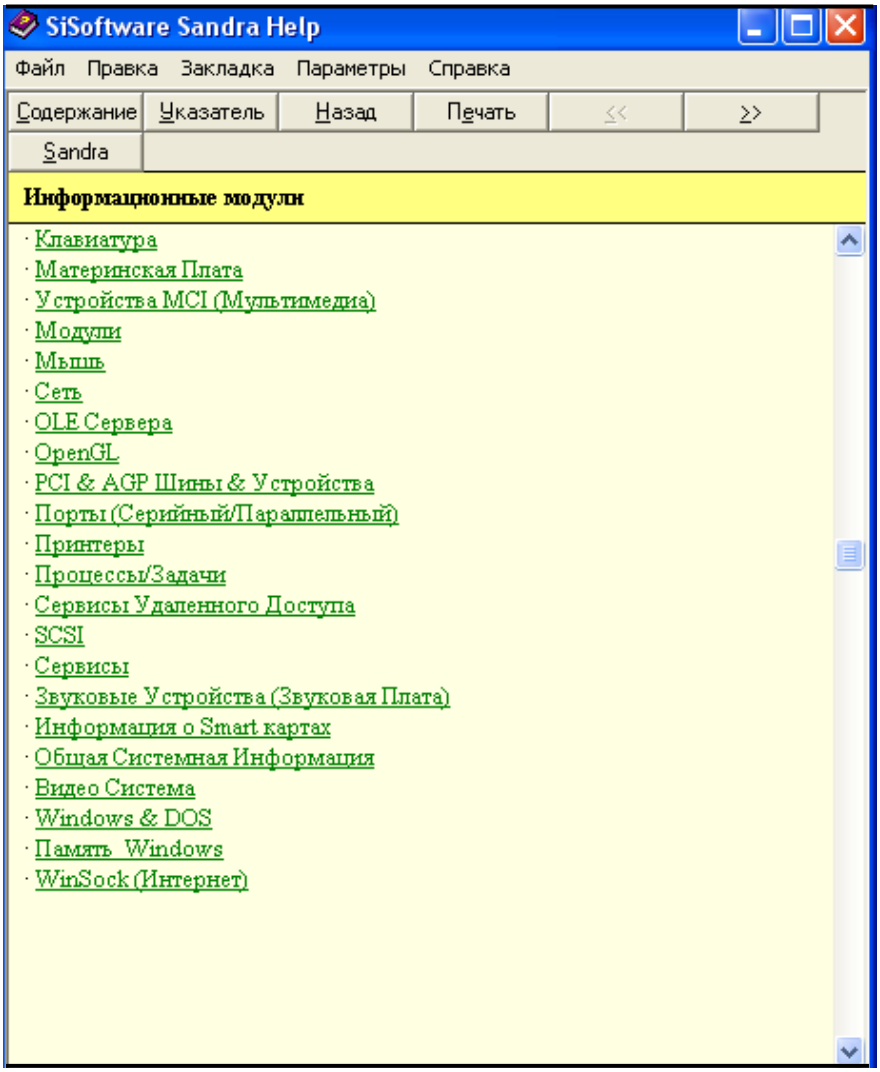


Рис. 10.4. Информационные модули программы Sandra

Модули тестирования программы Sandra (рис. 10.5) включают:

- тест CD-ROM/DVD;
- арифметический тест процессора;
- мультимедиа тест процессора;
- тест файловой системы;
- тест кэша и памяти (рис. 10.6);
- тест пропускной способности памяти (рис. 10.7);
- тест пропускной способности сети;
- видеотест;
- тест пропускной способности канала в Internet;
- тест переносного диска (Flash).

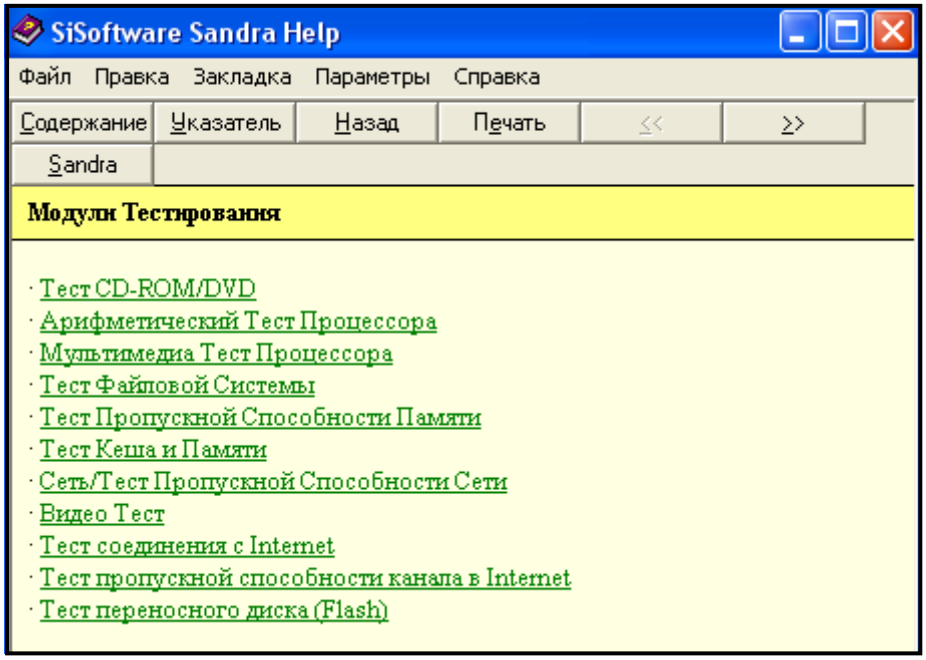


Рис. 10.5. Тестовые модули программы Sandra

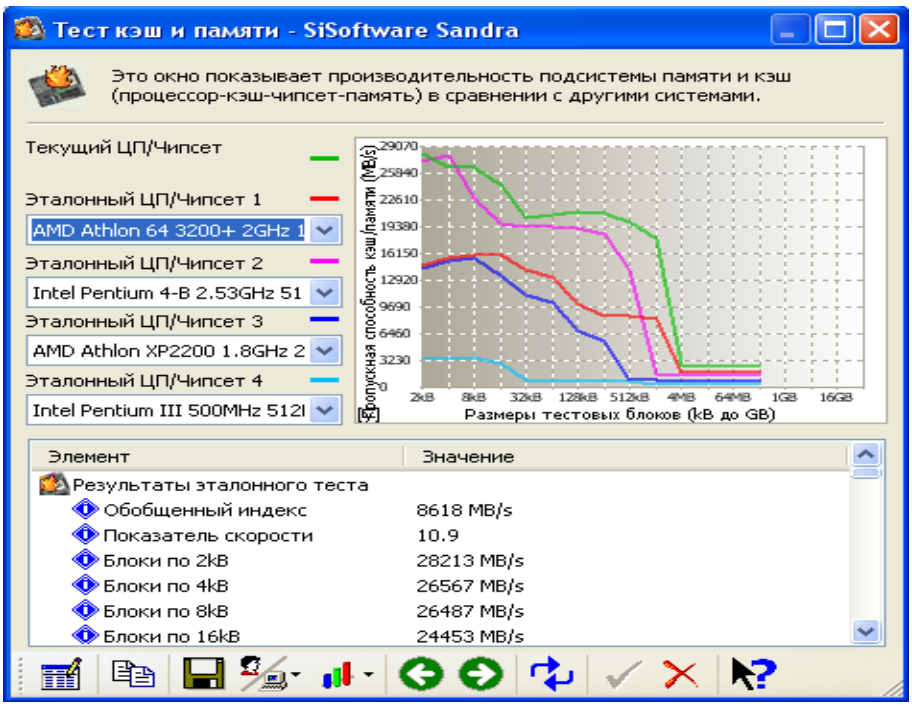


Рис. 10.6. Тест сравнения производительности процессора и подсистемы памяти с эталонными образцами

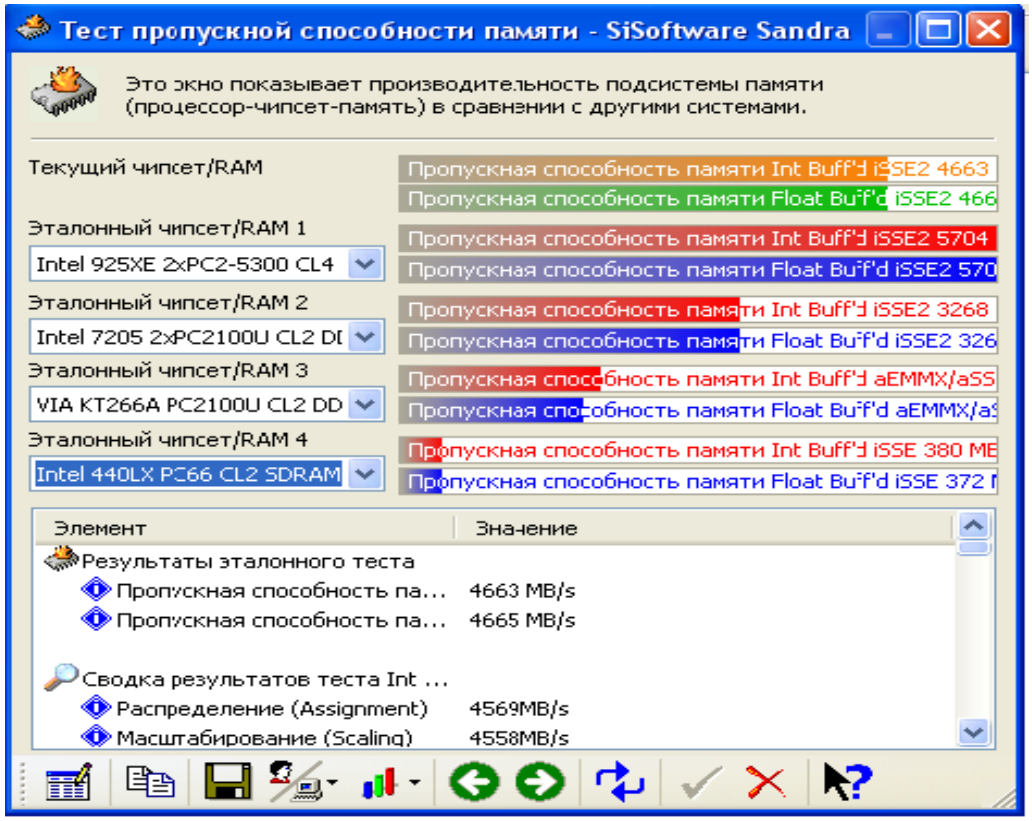


Рис. 10.7. Сравнение производительности подсистемы памяти с эталонными системами

Тест сравнения производительности процессора и памяти (тест кэш и памяти) позволяет выбрать в меню четыре эталонных процессора и произвести их графическое сравнение с исследуемым (нашим) процессором. С этой целью необходимо «щелкнуть» курсором по кнопке и получить график исследуемого процессора и сравнить его производительность.

Тестовая программа ASTRA

ASTRA (Advanced Sysinfo Tool and Reporting Assistant) – программа определения конфигурации и диагностики компьютера, позволяющая получить подробную информацию об аппаратном обеспечении компьютера и режимах его работы. Интерфейс программы ASTRA представлен на рисунке 10.8.

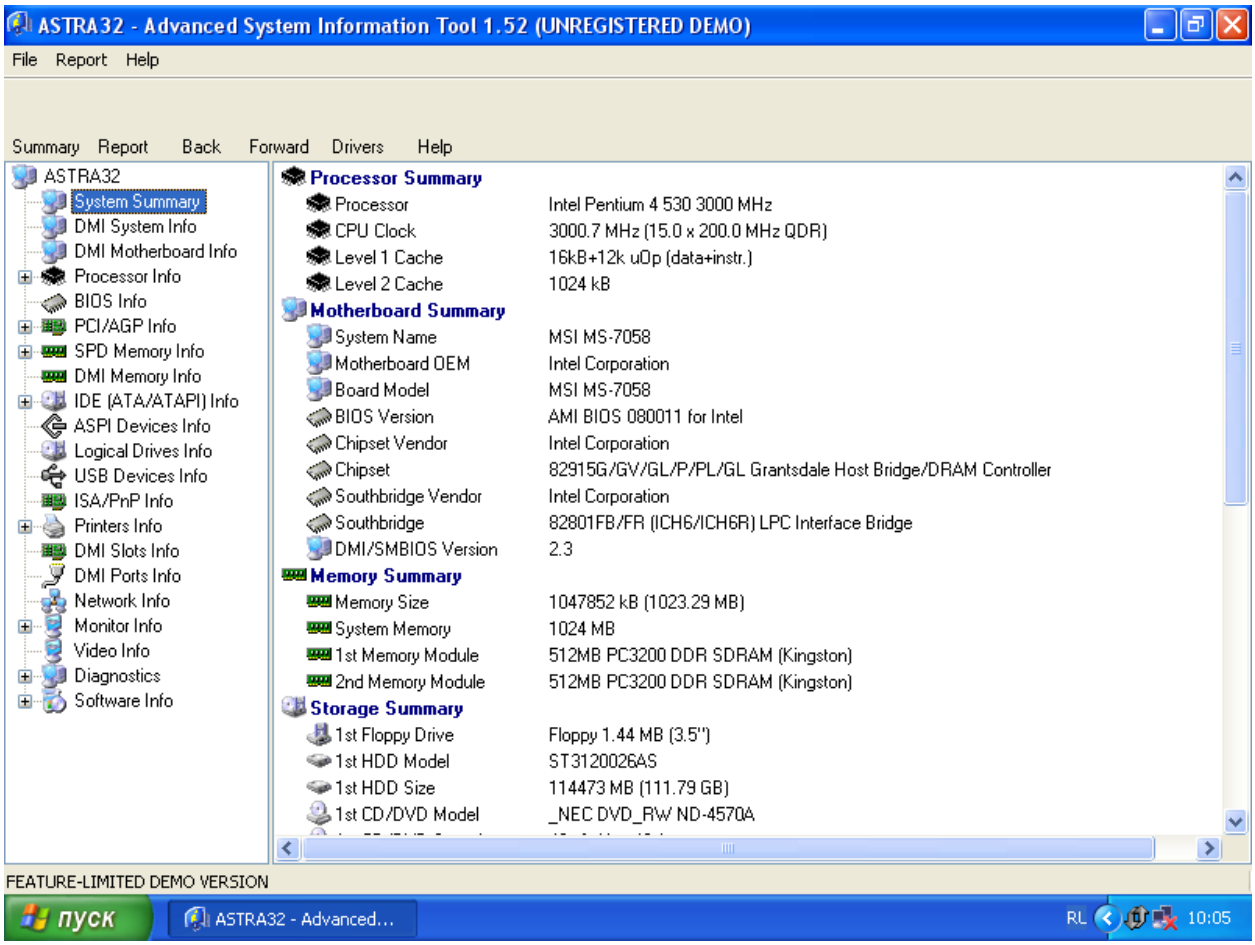


Рис. 10.8. Интерфейс программы Astra

Возможности тестовой программы:

– определение типа и тактовой частоты процессора, коэффициента умножения и частоты системной шины, типа разъема (slot, socket), определение поддерживаемых процессором возможностей, размера и параметров КЭШа;

– определение производителя материнской платы, BIOS и модели чипсета;

– определение надежности винчестеров с помощью уникального алгоритма;

– определение модели и емкости винчестера, CD/DVD устройства, типа интерфейса (Parallel ATA, Serial ATA I, Serial ATA II), а также скорости чтения/записи CD/DVD приводов;

– информация о модулях памяти (объем, тип, производитель, скоростные характеристики);

– определение производителя, типа и объема видеопамяти и поддерживаемых видеорежимов;

– определение типа звуковой карты, модема, сетевой карты, принтера, сканера;

– создание файла-отчета в текстовом INI, HTML, XML форматах;

– работа в режиме командной строки и запуска с загрузочной дискеты, CD/DVD диска или флеш накопителя;

– создание файла-отчета в формате программы инвентаризационного учета вычислительной техники.

Контрольные вопросы

1. Назовите назначение программы POST.
2. Назовите последовательность шагов POST.
3. На какие группы подразделяются тестовые программные средства?
4. Назначение программы Checkit.
5. Что позволяет определить System Info программы Checkit?
6. Назначение и возможности программы Sandra.
7. Какие модули входят в состав программы Sandra?
8. Назначение и возможности апрграммы Astra.

СПИСОК СОКРАЩЕНИЙ

АЛУ - арифметико-логическое устройство

АСУ - автоматизированная система управления

АЦП - аналого-цифровой преобразователь

БВР - блок внутренних регистров

БИС - большая интегральная схема

ВЗУ - внешние запоминающие устройства

ДС - дешифратор

ДТЛ - диодно-транзисторная логика

ДНФ - дизъюнктивная нормальная форма

ЕС ЭВМ - единая система ЭВМ

ЗУ - запоминающее устройство

ИМС - интегральная микросхема

ИС - интегральная схема

КМОП - транзисторная логика на комплементарных МОП-транзисторах

КНФ - конъюнктивная нормальная форма

КОП - код операции

КПД - контроллер прямого доступа к памяти

КУ - комбинационное устройство

ЛЭ - логический элемент

МОП - миталл-оксид-полупроводник

МП - микропроцессор

МПП - микропроцессорная память процессора

НГМД - накопитель на гибком магнитном диске

НЖМД - накопитель на жестком магнитном диске

НОД - накопитель на оптическом диске

НСТЛ - транзисторная логика с непосредственной связью

ОЗУ - оперативное запоминающее устройство

ОС - операционная система

ОШ - общая шина

ПЗУ - постоянное запоминающее устройство

ПК - преобразователь кодов

ПК - персональный компьютер

ПКН - преобразователь код-напряжение

ПНК - преобразователь напряжение-код

ПЭВМ - персональная ЭВМ

РТЛ - резисторно-транзисторная логика

CD - шифратор

СДНФ - совершенная ДНФ

СКНФ - совершенная КНФ

СМ - системная магистраль

СС - система счисления

ТПС - тестовые программные средства

ТТЛ - транзисторно-транзисторная логика

ТТЛШ - ТТЛ с диодом Шоттки

УВв - устройства ввода

УВВ - устройства ввода-вывода

УВыв - устройства вывода

УР - устройство равнозначности

УУ - устройство управления

ЦАП - цифро-аналоговый преобразователь

ША - шина адреса

ШД - шина данных

ШУ - шина управления

ЭВМ - электронно-вычислительная машина

ЭП - элемент памяти

ЭСЛ - эмиттерно-связанная логика

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Акулов О. А. Информатика: базовый курс: учебник для вузов / О. А. Акулов, Н. В. Медведев. - М.: Омега-Л, 2007.

2. Браммер Ю. А. Цифровые устройства: учебное пособие для вузов / Ю. А. Браммер, И. Н. Пащук. – М.: Высш. шк., 2004.

3. Бройдо В. Л. Вычислительные системы, сети и телекоммуникации: Учебник для вузов / В. Л. Бройдо. – 2-е изд. – СПб. : Питер, 2004.

4. Букреев И. Н. Микроэлектронные схемы цифровых устройств / И. Н. Букреев, В. И. Горячев, Б. М. Мансуров. – 3-е изд., перераб. и доп. – М.: Радио и связь, 1990.

5. Вычислительные системы, сети и телекоммуникации: учебник. – 2-е изд., перераб. и доп. / под ред. А. П. Пятибратова. – М.: Финансы и статистика, 2004.

6. Гилмор Ч. Введение в микропроцессорную технику / Ч. Гилмор. – пер. с англ. – М.: Мир, 1984.

7. Гитис Э. И. Аналого-цифровые преобразователи / Э. И. Гитис, Е. А. Пискулов. – М.: Энергоиздат, 1981.

8. ГОСТ 2.743–91. Обозначения условные графические в схемах. Элементы цифровой техники. – М.: Изд-во стандартов, 1993.

9. Гук М. Аппаратные средства IBM PC: энциклопедия / М. Гук. – СПб. : Питер Ком, 2004.

10. Жаров А. «Железо» IBM 2004 или все о современном компьютере / А. Жаров. – М.: «МикроАрт», 2003.

11. Захаров Н. Г. Цифровые и импульсные устройства: учебное пособие для вузов / Н. Г. Захаров. – Ульяновск : УлГТУ, 1997.

12. Информатика. Базовый курс / под ред. С. В. Симоновича. – 2-е издание. – СПб. : Питер, 2004.

13. Информатика: учебник / под ред. Н. В. Макаровой. – 3-е изд., перераб. – М.: Финансы и статистика, 2000.

14. Костин А. Е. Программное обеспечение микроЭВМ. В 11 кн. Кн. 1. Структура и функционирование микроЭВМ: учебное пособие для ПТУ / А. Е. Костин; под ред. В. Ф. Шаньгина. – М.: Высш. шк, 1991.

15. Петровский И. И. Логические ИС КР1533, 1534: справочник в 2-х частях / И. И. Петровский. – М.: Бином, 1993.

16. Могилев А. В. Информатика: учебное пособие / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. – М.: Академия, 2001.

17. Нарышкин А. К. Цифровые устройства и микропроцессоры: учеб пособие для вузов / А. К. Нарышкин. – М.: Издательский центр «Академия», 2006.

18. Никитин Б. Я. Теоретические основы вычислительной техники. Часть 2 / Б.Я. Никитин, В. Н. Скребков. – Л. : ЛВВИУС, 1988.

19. Новиков Ю. Персональные компьютеры / Ю. Новиков, А. Черепанов. – СПб. : Питер, 2001.

20. Острековский В. А. Информатика: учебник для вузов / В. А. Острековский – М.: Высшая школа, 2000.

21. Попов Л. Н. Цифровые функциональные устройства комбинационного типа: учебное пособие / Л. Н. Попов. – М.: Издательство МАИ, 1990.

22. Сайфутдинов Р. А. Принципы построения и функционирования компьютеров / Р. А. Сайфутдинов, А. Г. Назаров. – Ульяновск : УВВИУС, 2006.

23. Сайфутдинов Р. А. Вычислительная техника и информационные технологии. Часть 1 / Р. А. Сайфутдинов, А. Г. Назаров, В. А. Лукьянов. – Ульяновск : УВВИУС, 2007.

24. Соломенчук В. Г. Аппаратные средства персональных компьютеров / В. Г. Соломенчук. – СПб. : БХВ-Петербург, 2003.

25. Стрыгин В. В. Основы вычислительной микропроцессорной техники и программирования / В. В. Стрыгин, Л. С. Щарев. – М.: Высшая школа, 1989.

26. Таненбаум Э. Архитектура компьютера / Э. Таненбаум. – СПб. : Питер, 2004.

27. Угрюмов Е.П. Цифровая схемотехника / Е. П. Угрюмов. – СПб. : БХВ-Петербург, 2002.

28. Цифровая и вычислительная техника: учебник для вузов / Э. В. Евреинов, Ю. Т. Бутыльский, И. А. Мамзелев и др. / под ред. Э. В. Евреинова. – М.: Радио и связь, 1991.